



**INSTITUT FÜR
ENERGIETECHNIK
UND THERMODYNAMIK**

Institute of Energy Systems and Thermodynamics

Master's Thesis

Optimizing the efficiency of hydropower plants with Artificial Neuronal Networks

under the supervision of

Univ.Assoc. Dipl.-Ing. Bernhard Semlitsch, PhD

E302 - Institute of Energy Systems and Thermodynamics

submitted to the Faculty of Mechanical and Industrial Engineering
of Technische Universität Wien
for the degree of Diplom-Ingenieur (Dipl.-Ing.)

by

Wenzel Müller, BSc

Matr.Nr. 01604798

Laudongasse 25/16, A-1080 Vienna

Statutory Declaration

This thesis is the result of my own work and includes nothing that is the outcome of work done in collaboration except as specified in the text.

It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at Technische Universität Wien or any other University or similar institution except as specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at Technische Universität Wien or any other University or similar institution except as specified in the text.

Vienna, July 2025



(Wenzel Müller, BSc)

Acknowledgements

I would like to extend my sincere gratitude to Bernhard Semlitsch for his invaluable supervision throughout this thesis. His expertise, and prompt feedback were crucial for ensuring the steady and coherent progress of this work.

This thesis would not have been possible without the initiative and continuous support of Christoph Libisch-Lehner, who not only envisioned the original idea but also played a key role in the development of the program. His creativity in problem-solving and his thorough reviews of both concepts and code significantly contributed to the quality and outcomes of this research.

I am especially thankful to Thomas Weissensteiner and Philip Kotrba for motivating the collaboration between the Technical University of Vienna and AFRY Austria. Their encouragement and support laid the foundation for this thesis to take shape.

Special thanks go to all the friends I met during my studies. Their support, both professionally as well as personally, was invaluable. Additionally, I am grateful for the lasting friendships that have grown from this journey.

Finally, I wish to thank my family and girlfriend for making this academic path possible. Their emotional support and encouragement allowed to overcome every challenge and pursue my goals with confidence.

Abstract

The motivation for this work is an existing computer program developed by AFRY Austria. The program receives hydrological inflow time series and can simulate electricity generation based on the water flow distribution and hydro power plant characteristics. The aim of this work is to improve the modelling of the electro-mechanical equipment by solving the optimisation task of optimal load distribution in a hydro power plant with an arbitrary number of machine sets. For this purpose, a computer program was developed, which creates a Radial Basis Function Neural Network (RBFNN) and optimises its parameters with the NSGAI algorithm.

The work consists of the mathematical formulation of the objective function for the underlying problem, the development of an RBFNN and the optimisation of the network parameters with the NSGAI algorithm. A special RBFNN structure is proposed and adapted to the problem. An equation for estimating a proper network size according to the simulated power plant was developed. For validating the results, a second algorithm was developed. It finds the optimal solution to the problem by brute force, meaning that it calculates every possible solution and selects the optimum. The results of both programs are compared and analysed.

The results confirmed that the RBFNN is capable of approximating the objective function. However, compared to the validation method, the RBFNN's output could not reach the optimum of the objective function. For tasks with several power units of different characteristics, it was not possible to optimise the RBFNN's parameters to achieve satisfying results. The work concludes that the RBFNN can be used in a machine learning task for the same problem, but the method using an optimiser for finding the optimal RBFNN parameters proves to be computationally inefficient, while lacking transparency of the solution process.

Kurzfassung

Als Grundlage dieser Arbeit dient ein Computerprogramm der Firma AFRY Austria. Das Programm erhält hydrologische Zuflusszeitreihen und kann auf Grundlage der Wasserbilanz die Stromerzeugung eines Wasserkraftwerkes simulieren. Das Ziel dieser Arbeit ist es die Modellierung der elektromechanischen Ausrüstung zu verbessern, indem das Optimierungsproblem der optimalen Lastverteilung auf eine beliebig wählbare Anzahl an Maschinensätzen im Kraftwerk gelöst wird. Zu diesem Zweck wurde ein Neuronales Netz mit radialen Basisfunktionen erstellt und dessen Parameter mit dem NSGAI-Algorithmus optimiert. Die Arbeit besteht aus der Formulierung der Zielfunktion für das vorliegende Problem, der Entwicklung des aus Radialen Basis Funktionen bestehenden Neuronalen Netzwerkes (RBFNN) und der Optimierung der Parameter unter Verwendung des NSGAI-Algorithmus. Es wird eine angepasste Netzwerkstruktur vorgeschlagen und eine Gleichung zur optimalen Auswahl der Netzwerkgröße präsentiert, welche im Laufe der Arbeit entwickelt wurde. Um die Ergebnisse validieren zu können, wurde ein zweiter Algorithmus entwickelt. Dieser berechnet die optimale Lösung durch die Methode der rohen Gewalt, das bedeutet es werden alle möglichen Lösungen berechnet und anschließend der optimale Lösungspfad ausgewählt. Die Ergebnisse der beiden Programme werden verglichen und analysiert. Die Ergebnisse bestätigen die Fähigkeit des RBFNN die Zielfunktion zu approximieren. Im Vergleich zur Validierungsmethode konnte das Ergebnis des RBFNN jedoch nicht das Optimum mit der gleichen Genauigkeit bestimmen. Bei komplexeren Aufgaben war es nicht möglich die Parameter des RBFNN ausreichend zu optimieren, um zufriedenstellende Ergebnisse zu erhalten. Die Arbeit diskutiert, dass das entwickelte RBFNN Modell für maschinelles Lernen verwendet werden könnte. Für die aktuelle Anwendung mit einem Optimierer zur Suche der optimalen RBFNN-Parameter, erweist sich die Methode auf Grund des hohen Rechenaufwandes und der mangelnden Transparenz der Berechnung als ineffizient.

Contents

1 Introduction	1
1.1 Motivation	1
1.2 Scope of work	2
2 Theory	4
2.1 Hydropower Plants	4
2.1.1 General	4
2.1.2 Mathematical formulation	5
2.2 Artificial Neural Networks	9
2.2.1 General	9
2.2.2 Radial Basis Function Neural Networks	12
2.3 Genetic algorithms	15
2.3.1 General	15
2.3.2 Non-Dominant Sorting Genetic Algorithm II	16
3 System description	17
3.1 Problem definition	17
3.2 Code structure	20
3.2.1 Optimisation mode	20
3.2.2 Parameter evaluation	26
3.2.3 Operation mode	28

4 Results	30
4.1 Basic example	30
4.1.1 System Validation	34
4.2 Setting system parameters	38
4.2.1 Optimal number of nodes	38
4.2.2 Finding optimal settings for NSGAI	41
4.3 Four unit example	43
5 Discussion	46
5.1 Validation of the RBFNN results	47
5.2 Future works	48
6 Conclusion	50
Bibliography	52
A Tables	55
A.1 Basic example	55
A.1.1 Input files	55
A.1.2 Output files	64
A.2 Four unit example	81
A.2.1 Input files	81

Nomenclature

Abbreviations

NSGAI	Non-dominant sorting algorithm II
Pop_size	Population size of the NSGAI algorithm
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network

Physical Constants

e	Euler's number	-
g	Gravitational acceleration	m/s ²

Latin Symbols

A	Area	m ²
d	Diameter	m
f	Output function	-
$f()$	Function of ()	-
H	Waterhead	m
h	head losses	m
I	Current	A
l	Pipe length	m
P	Power	W
p	Pressure	P
Q	Volume flow rate	m ³ /s
r	Output RBFNN	-
U	Voltage	V
X	Node input	-
x	Load factor	-
Y	Node output	-

z	Height	m
-----	--------	---

Greek Symbols

$\cos \phi$	Power factor	-
η	Efficiency	-
λ	Friction loss coefficient	-
μ	Center of RBF	-
ω	Node-weight	-
ω	Rotational speed	s ²
Φ	Radial basis function	-
ρ	Density	kg/m ³
Θ	Activation function	-
θ	Node threshold	-
θ	Radial basis function width	-
ζ	Minor loss coefficient	-

Vectors and Tensors

$\vec{\theta}$	Vector of radial basis function width
\vec{X}	Vector of neural network input

Chapter 1

Introduction

1.1 Motivation

The foundation for this work is an existing computer program simulating hydropower plants, which was developed by AFRY Austria. The program can simulate all types of hydropower plants, such as storage, diversion, run-of-river and pumped storage power plants, based on their technical characteristics.

The technical characteristics describe the hydraulic (reservoir, headrace) and electro-mechanical (turbine, generator, transformer) components of planned or existing hydropower plants. An application is to provide the program with hydrological inflow time series to dynamically simulate energy generation based on the water balance and power equation for each timestep. The decisive factor is how much water is to be released by the turbines at a particular time. This is determined using optimisation methods (machine learning) and framework conditions, which allow for target functions to be defined. For water management issues, AFRY's program usually forecasts daily behaviour. Therefore, the electro-mechanical data was simplified. The E&M module of the program is now to be improved in order to better answer energy management questions and increase temporal accuracy.

The aim of this work is to consider the electro-mechanical components in more detail. It should be possible to define the type of turbine, the number of turbines, and the generator, including the actual efficiencies as well as the load distribution between the hydropower units. The program is developed further to fulfil these requirements and amplify the capabilities of the existing program.

The mathematical description of a hydropower unit and the corresponding electrical equipment is well-studied. Adding several units together in one power plant creates a non-trivial function. Many approaches for optimising this function exist [24]. This thesis creates the foundation for introducing a new approach, which uses artificial neural networks. It offers the unique possibility to provide a continuous function as output. This characteristic might offer the possibility of real-life applications once the parameters of the network are optimised.

1.2 Scope of work

Based on the aforementioned motivational statements, the research question of this thesis is defined as:

Can an artificial neural network be used to simulate the optimum load distribution in a hydropower plant with an arbitrary number of machine sets?

The main goal of the thesis is to develop a neural network capable of simulating various types of hydropower plants and optimising their production by selecting the optimal volume flow distribution on their power units. Therefore, this thesis focuses on the following key aspects:

1. Selection of a suitable neural network
2. Mathematical description of the problem
 - (a) Mathematical description of the hydraulic machine(s)
 - (b) Mathematical description of the generator(s)
 - (c) Mathematical description of the transformer(s)
 - (d) Definition of the boundary condition of all components
3. Optimising the neural network
4. Validation of the results

The original intent was to develop an artificial neural network which gets trained by real-world data. However, it became apparent that this task would be too extensive. AFRY is an engineering consulting company which is often involved in

the tendering process of new hydro power plant projects. For this reason, a program which is able to simulate a future power plant, without the need for real-world data, is far more beneficial. Due to those two reasons, the scope was altered to optimising a neural network with the help of a numerical optimiser, instead of training it with training data. This approach still allows for future development of a machine learning application with the same neural network.

It should be highlighted that the developed program does not have a time discretisation. However, in connection with AFRY's program, mentioned in [1.1](#), it is used for simulating power generation on basis of time steps, which can reach from hourly to every 15 minutes, allowing the problem to be limited to quasi-steady operating conditions. For this reason, it is not necessary to investigate transient flow conditions. The developed program finds the best flow distribution just based on the total inflow and the provided system parameters. The time steps of AFRY's program do not serve as input to the newly developed program.

Chapter 2

Theory

The theoretical background of this work can be categorised into three fields: Hydropower plants, Neural Networks, and Optimisation Algorithms. These categories are also used to structure this chapter. Because all topics could be described in detail, only the theory which was directly used in this thesis is described.

2.1 Hydropower Plants

2.1.1 General

Hydroelectric power uses the kinetic energy of water to produce electricity [12]. Four main types of hydropower plants exist. Run-of-river hydropower uses the natural flow of a river for power generation. Storage hydropower also uses natural inflow, while storing it for later usage in a reservoir. Pumped storage hydropower pumps water from a lower reservoir to a higher reservoir. It stores the energy supplied by pumping as potential energy and is able to convert the energy back at a later point. Finally, there is also offshore hydropower, which is getting more and more established in recent years [2]. These different kinds of hydropower can overlap or be combined in one project. For example, run-of-river projects have some amount of storage capability.

Hydropower plants have to be adapted to the local conditions. The basic components are always the same and are listed in [2.1]. From an electromechanical point of view, the intake, penstock, hydraulic machine, draft tube, generator and

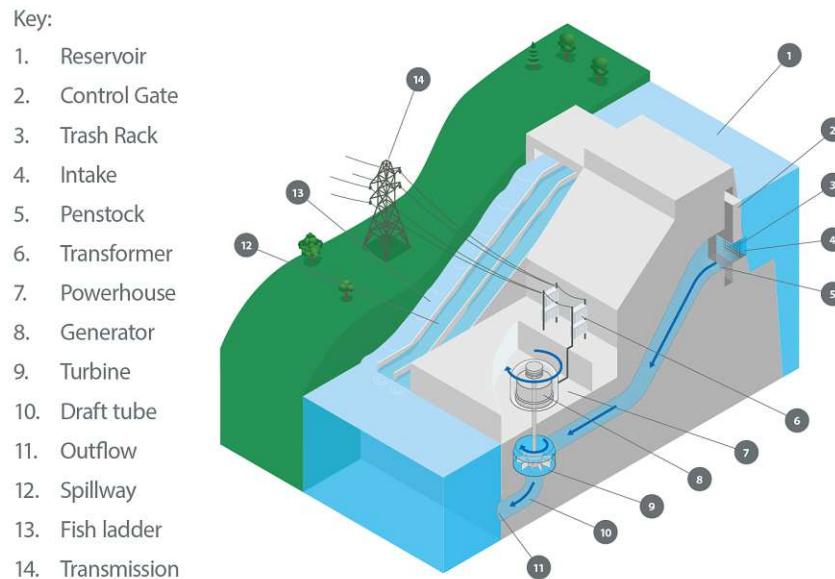


Figure 2.1: Components of a hydropower plant [2]

transformer are of the highest importance because they highly contribute to the overall efficiency of a power plant.

2.1.2 Mathematical formulation

Hydraulic machine power and efficiency

The power production of a hydraulic machine can be calculated by the equation

$$P = \eta_h \rho g H \dot{Q} \quad (2.1)$$

where ρ is the water density, g is the gravitational constant, H is the water head, and \dot{Q} the volume flow rate [30]. Equation 2.1 shows that the available energy depends on the potential energy of the water, which is a function of the upper and lower water levels (water head), and the volume of water. The power production is limited by the rate at which this water can be released. The hydraulic machine also has an efficiency, η_h . It depends on the volume flow rate, water head, and rotational speed of the hydraulic machine runner. It can be obtained through hill chart measurements. The efficiency hill chart shows the efficiency as a function of

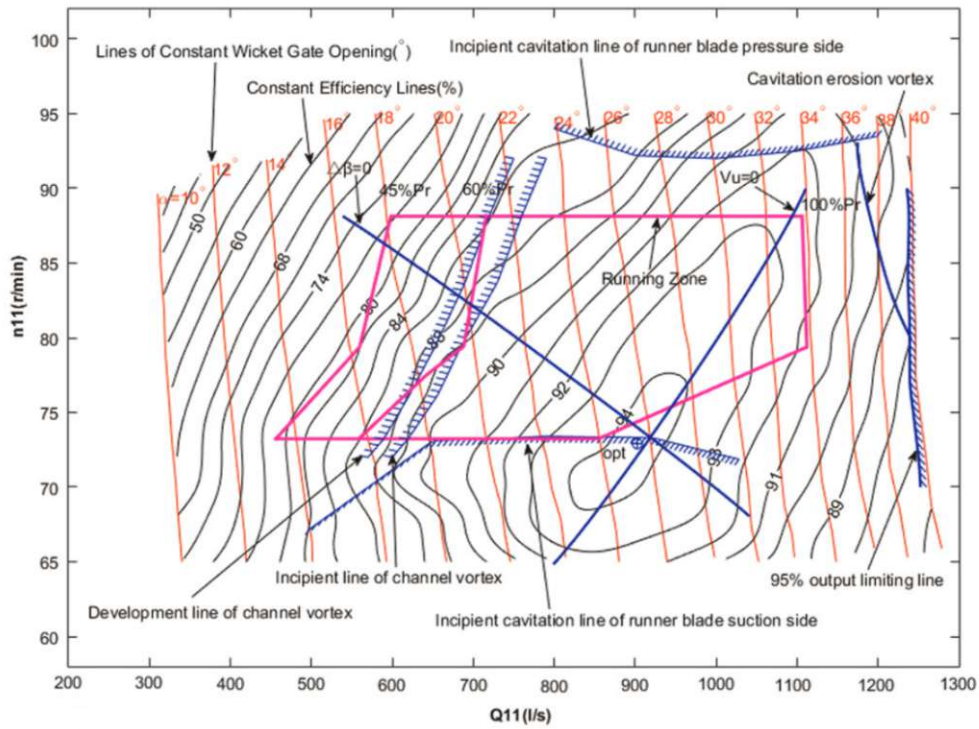


Figure 2.2: Efficiency hill chart [14]

the rotational speed and volume flow rate at a certain water head, which can be expressed as,

$$\eta_h = \eta_h(\omega, H, \dot{Q}). \quad (2.2)$$

Efficiency hill charts of hydraulic machines are typically created by using characteristic model runners due to the high effort and related costs. The test should cover the entire operating range of a hydraulic machine. The result is a table with operating points, which can be displayed as a hill chart. The rotational speed is the ordinate, and the volume flow is the abscissa. Operating points with the same efficiency are connected through interpolated lines. Additional information like guide-vane angles curves, output limiting line, and cavitation curves are added to hill charts, as shown in Figure 2.2 [14].

Penstock head losses

The head losses of a penstock can be categorised into friction losses h_f and minor losses h_r . The friction losses result from the fluid's viscosity. For a laminar flow, the frictional losses can be written as

$$h_f = \lambda \frac{l}{d} \frac{v^2}{2g} \quad (2.3)$$

which is referred to as the Darcy–Weisbach equation. Minor losses occur in pipe installations like elbows, valves, or manifolds. They are a result of energy dissipation due to mixing processes in turbulent flows. Minor local losses can be described as the product of a minor loss coefficient ζ and the flow velocity head

$$h_r = \zeta_1 \frac{v_1^2}{2g}. \quad (2.4)$$

The sum of both head loss types

$$h_l = \sum h_f + \sum h_r \quad (2.5)$$

is the total head loss of a penstock. As visible in equations [2.3](#) and [2.4](#), frictional head losses as well as minor head losses both depend on the flow velocity squared and therefore on the volume flow rate squared. [28](#)

The total head losses of an existing penstock can be measured during its operation. To perform this calculation, two specific points along the penstock must be defined. The required measurements include static pressure, p , volumetric flow rate, \dot{Q} , cross-sectional area, A , and the elevation difference between these two points, Δz . Using these parameters, the head loss of the penstock can be computed using the following equation derived from the Bernoulli equation [32](#):

$$h_l = \left(\frac{p_1}{\rho g} + \frac{\dot{Q}_1^2}{2gA^2} + z_1 \right) - \left(\frac{p_2}{\rho g} + \frac{\dot{Q}_2^2}{2gA^2} + z_2 \right) \quad (2.6)$$

As a simplification, the head losses of the whole penstock can be described in one loss coefficient. That way, the penstock losses at specific volume flow rates can be approximated.

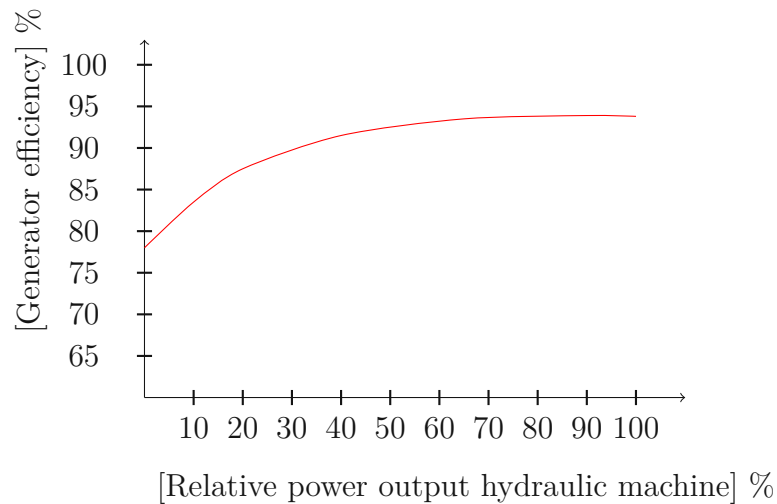


Figure 2.3: Schematic reproduction of a synchronous generator efficiency curve adapted from [18]

Generator

The generator in a hydropower plant is attached to the hydraulic machine's shaft and converts the mechanical energy to electrical energy. Typically, synchronous or asynchronous generators are used. The latter is mainly used for small hydro (less than 5 MW) due to their lower cost [13]. Generator losses can be classified into electrical losses and mechanical losses. The main losses are copper, iron, core, rotational, and friction losses. [29] The generator efficiency is a function of the load factor, x , and the power factor, $\cos\theta$ [19]

$$\eta_g = f(x, \cos\theta) . \quad (2.7)$$

The typical efficiency curve of a generator is shown in Figure 2.3, where the efficiency η_g is plotted over the power output [18].

Pump storage hydropower plants using pump turbines use their generator also as a motor. The basic design is the same as for a machine just used as a generator. Generator motors just need additional equipment for start-up. [13]

Transformer

A transformer is a stationary electrical machine used in power transmission to transfer electrical energy between circuits. It functions as a voltage regulation device commonly applied in AC power distribution and transmission systems. Transformers are designed to step-up or step-down the AC voltage while maintaining the same frequency by establishing a conductive link between circuits. They are particularly used between the primary distribution circuits and the power generator. They play a crucial role in distribution networks. [7]

Similar to generators, transformers have copper, iron, and core losses. Copper losses are induced by the ohmic resistance in the copper coils. These are quadratically proportional to the current. The iron and core losses are caused by the reversal of magnetism, which is quadratically proportional to the voltage and the frequency. Additional iron core losses are induced by eddy currents in the iron core. Similar to the copper losses, the eddy currents are induced by the ohmic resistance and, therefore, are quadratically proportional to the current. [3]. Thus, the transformer efficiency can be described as,

$$\eta_t = \eta_t(U^2, I^2, f, \cos\phi). \quad (2.8)$$

In figure [2.4], the relative partial load losses and efficiency curves are plotted, showing that the transformer has a high efficiency which remains constant over a wide range of partial load.

2.2 Artificial Neural Networks

2.2.1 General

An Artificial Neural Network (ANN) comprises an arbitrary number of interconnected nodes. One node is the smallest computational component of an ANN. The most common type of node is the "threshold" node, which can be mathematically described as,

$$Y_i = f\left(\sum_j w_{i,j} X_j - \theta_i\right) \quad (2.9)$$

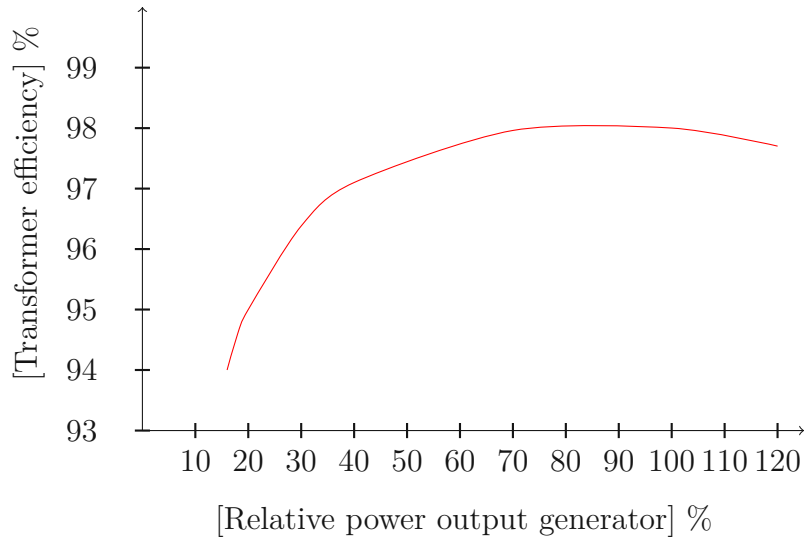


Figure 2.4: Schematic reproduction of a transformer efficiency curve adapted from [3]

where X_j are the inputs from preceding nodes, $w_{i,j}$ are the corresponding weights from nodes j to node i and Y_i is the summed output. [26] The basic structure of a node is shown in Figure 2.5. When the accumulated sum of the weighted input $X_j w_j$ reaches the threshold θ_i , the node is activated and sends its output to the following nodes. The function, f , is called the activation function.

Activation functions

There exist many different activation functions. One of the first ANNs proposed by McCulloch and Pitts used a simple step function [23],

$$\Theta(X; \theta_i) = \begin{cases} 0 & \sum_j (X_j w_{i,j}) > 1 \\ 1 & \sum_j (X_j w_{i,j}) < 1 \end{cases}. \quad (2.10)$$

Another common activation function is the sigmoid-shaped function. [13] It was first proposed by Rumelhart et al. in 1986 and is defined as [14]

$$\Theta(X; \theta_i) = \frac{1}{1 + e^{-\sum_j (X_j w_{i,j} - \theta_i)}} \quad (2.11)$$

The sigmoid-shaped activation function is often used because of its nonlinearity

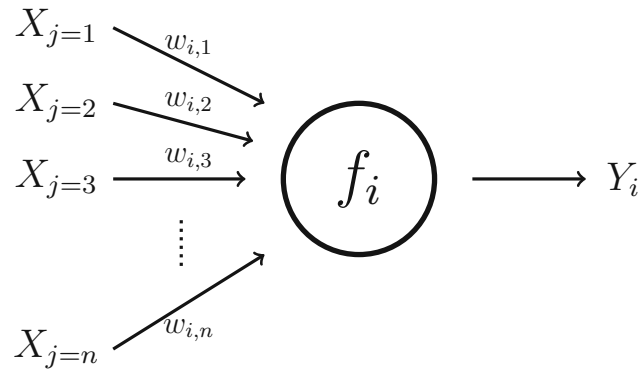


Figure 2.5: Single node of an ANN.

and differentiability. Without the nonlinear property of the activation function, an ANN could not perform nonlinear transformations. Depending on the studied problem, nonlinearity is essential for the network to accomplish its target function. The differentiability is required to output continuous functions [25](#).

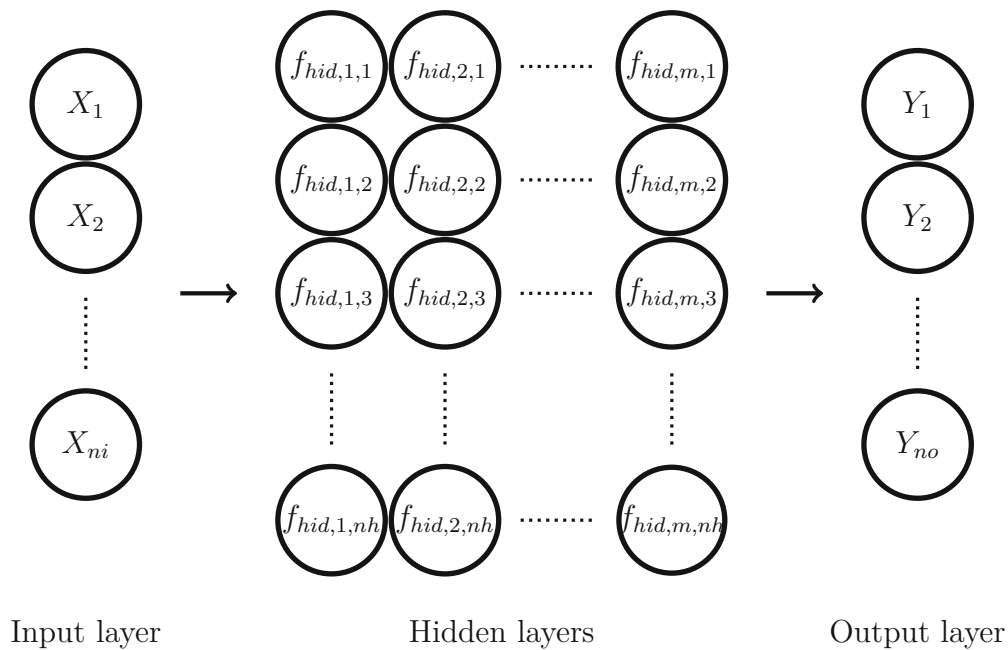


Figure 2.6: Topology of an ANN

General Topology

The nodes of ANNs are organised as layers. Typically, an ANN has one input layer and one output layer. Additionally, an ANN can have none or several hidden layers in between, as shown in Figure 2.6. The number of nodes and layers depends highly on the type of network and the complexity of the problem to be solved [33].

Node connectivity

An ANN can be classified into feed-forward and feed-backward networks depending on its node connections. Feed-forward networks can only send signals in one direction between layers. The second type, feed-backward networks, can also send information from a subsequent layer to a preceding layer [33]. These different types of connections result in fundamental differences between networks. Feed-forward networks associate one output with one input, whereas feed-backward networks produce many outputs for the same input until they reach an equilibrium during training.

Training methods

ANNs can be trained with different methods. Typically, supervised or unsupervised learning is applied. To successfully train a network, a dataset representing the whole feature space is required. In the case of the data instances being labelled, the training method is referred to as supervised learning. The learning algorithm changes the network parameters depending on the error between the labelled data and the network outputs [20]. Training algorithms, which do not utilise training data labels, are classified as unsupervised learning methods [11].

2.2.2 Radial Basis Function Neural Networks

In this thesis, a Radial Basis Function Neural Network (RBFNN) was chosen, because it was already successfully used to optimize water resources systems [21]. The basis for such networks was led by Powell in 1987, who used Radial Basis Functions (RBFs) for interpolation in high-dimensional spaces [27]. This technique was further developed into a multilayer network, which today is called RBFNN.

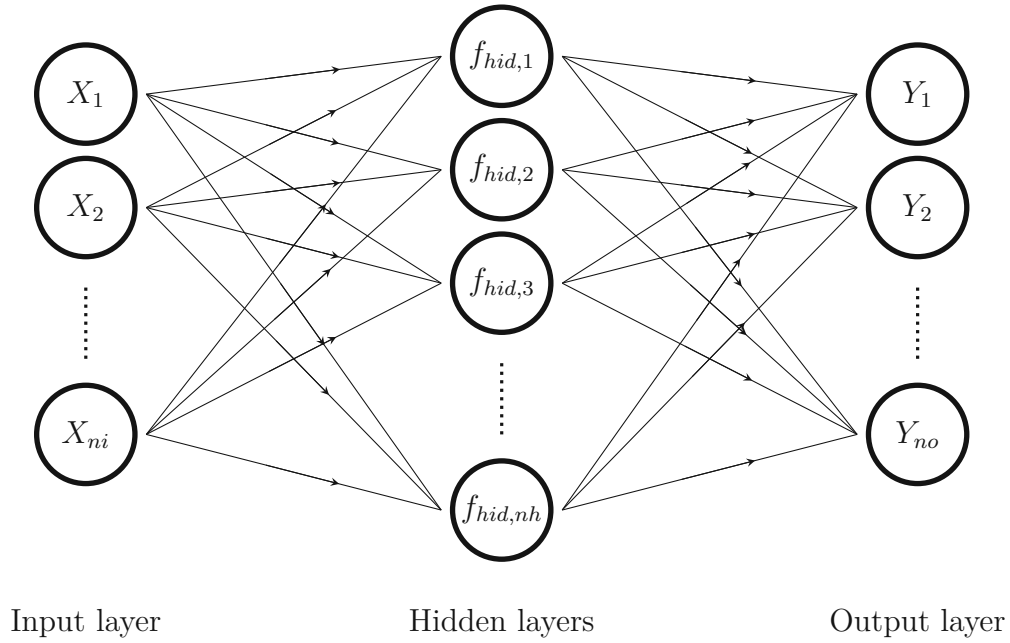


Figure 2.7: Topology of an RBFNN

This type of ANN consists of an input layer, a hidden layer and an output layer as visualised in figure 2.7 [22]. The vectors between the layers show that the RBFNN is a feed-forward network as described in 2.2.1.

The basic RBFN with n -hidden nodes and a single output can be formulated as

$$f(\vec{X}) = \sum_{i=0}^n \omega_i \Phi_i(\|\vec{X} - \vec{\mu}_i\|, \vec{\theta}_i) \quad (2.12)$$

where $\vec{X} \in \mathbf{R}$ is the input vector, ω_i are the weights of the network, and Φ_i are the RBFs. There exist many different RBFs. In general, the same RBF is used for all hidden nodes. In equation 2.12, $\vec{\mu}_i$ describes the centers of the RBFs, $\vec{\theta}_i$ describe the width of the RBFs and $\|\cdot\|$ denotes the Euclidean norm [8].

RBFNNs are well-suited for universal approximation tasks. Theoretically, they are able to approximate any continuous function as long as the number of hidden nodes is chosen accordingly and the centres and widths are adjusted to the specific task. The chosen function has to be continuous, integrable and locally bounded. [31] A few examples of typical RBFs are provided in table 2.1. The approximation capability is not dependent on the type of RBF. [31]

The activation function plays an important role in an RBFNN. As mentioned

RBF type	Mathematical formulation
Gaussian function	$\Phi(X) = \exp\left(-\frac{\ X - \mu\ ^2}{\theta^2}\right)$
Inverse multi-quadric function	$\Phi(X) = (\ X - \mu_i\ ^2 + \theta^2)^{-1/2}$
Multi-quadric function	$\Phi(X) = (\ X - \mu_i\ ^2 + \theta^2)^{1/2}$
Thin-plate-spline function	$\Phi(X) = X^2 \ln(X)$

Table 2.1: Typical radial basis functions [1]

in Chapter 2.2.1, activation functions are necessary for approximating nonlinear functions. For RBFNNs, the type of activation function combined with randomly generated hidden nodes does not influence their convergence capabilities as long as they fulfil the following requirements: The activation function has to be an unbounded non-constant piecewise continuous function $g : \mathbf{R} \rightarrow \mathbf{R}$. [17] The study also claims that the RBFs of the hidden nodes can be randomly initiated, although this theory is not proven. A brief overview of the advantages and disadvantages of RBFNNs from the literature is presented below.

RBFNNs Advantages

- Simple architecture: Their single hidden layer makes their design and interpretation far simpler compared to other types of ANNs [31].
- Good local generalization capabilities: Changes in the input space only effect nearby outputs [31].
- Great approximation capabilities: They can approximate any continuous function as long as a sufficient number of nodes is provided [31].
- Interchangeable RBFs: Different types of RBF can be used, appropriate to different tasks [31]

RBFNNs Disadvantages

- Poor approximation capabilities for constant functions: They can also approximate total or partially constant functions, but there is a trade-off between precision and local sensitivity [8].

- Scalability: For a high number of decision variables many nodes are needed requiring a high computational effort [8].

2.3 Genetic algorithms

2.3.1 General

One type of optimisation algorithm are genetic algorithms, which have analogies to evolution in biology. The two main features of evolution are:

- Inheritance and
- Mutation.

Inheritance is the transfer of features from one generation to the next, while not remaining identical. Mutation describes that over time, a major, unpredictable change occurs. Most of these easily noticeable changes are lethal and thus cannot be sustained over time. However, some mutations result in a significant advantage over previous generations and thus spread quickly in preceding generations. By taking advantage of those two characteristics, genetic algorithms follow the basic scheme presented below [15]:

1. Create and evaluate the initial population
2. Select a pair of solutions and create new ones by a process named *cross over*
3. Include some major changes to simulate mutation
4. Evaluate new solutions
5. Add the solution gained by *crossover* to the new population
6. If the new population has fewer members than the original one, repeat from step two onward and otherwise, continue to the next step
7. Keep new population and start again at step two, unless stopping criteria are met, then terminate process

All steps above, despite evaluation, are driven by random number generators. Therefore, generic algorithms are part of stochastic optimisation methods.

2.3.2 Non-Dominant Sorting Genetic Algorithm II

The Non-Dominant Sorting Genetic Algorithm II (NSGAI) algorithm used in this thesis was introduced by researchers from the Indian Institute of Technology Kanpur. The NSGAI is a multi-objective evolutionary algorithm which can find multiple Pareto-optimal solutions in a single run. Compared to similar algorithms, the NSGAI is less computationally expensive, can maintain diversity among obtained solutions, has a fast sorting procedure, and offers a parameter-less approach [9].

Chapter 3

System description

In this chapter, the computer program, which was implemented in Python, is described. The system comprises the neural network, the mathematical description of the the electro-mechanical hydropower plant components, and the NSGAI algorithm. The nomenclature was adjusted for this and the following chapters to highlight the change from the general description to the application. The RBFNN input is changed from X to Q , and the output from Y to r . Because the output of the RBFNN is passed through an activation function and a post-processing procedure, the output gets transformed from r to $r_{softmax}$ to r_{dist} .

3.1 Problem definition

The goal of the computer program is to optimise the water flow distribution in a hydropower plant with more than one hydraulic machine based on the hydrological inflow time series and the mechanical and electrical characteristics of this power plant. These characteristics have to be defined a priori and can be taken from a real or a fictitious power plant. To limit the complexity of the problem, some simplifications were made, which will be described in this chapter. The required input comprises:

1. Physical constants and parameters, as stated in [A.2](#)
2. Efficiency data of the hydraulic machine [A.1](#)

$Unit_i$ with $i=1,2,\dots,N$

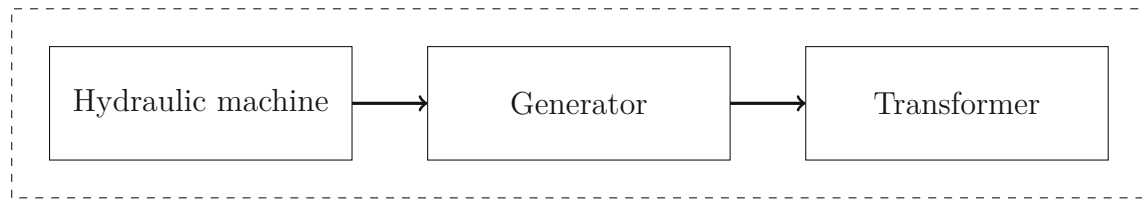


Figure 3.1: Single unit layout

3. Operational limits of the hydraulic machine
4. Efficiency data of the generator, as stated in [A.4](#)
5. Efficiency data of the transformer, as stated in [A.9](#)

The reason for this selection of input data is presented in the following paragraphs. The model output is a parameter set of an RBFNN. The network is capable of returning the optimal flow distribution by receiving a discretised volume flow rate and a fixed water head. Additionally, it saves a table which lists all outputs of the network depending on the discretised input vector of volume flows and the water head. It, therefore, displays the discretised solution space of the network. An example of the output can be found in the appendix [A.3](#).

Basic layout of the hydropower plant

The components of a hydropower plant are described in Section [2.1](#). The computer program only allows for a serial combination of one hydraulic machine with one generator and one transformer. This kind of arrangement is state-of-the-art in hydropower. No real-life examples of connecting several generators to one transformer or combining different numbers of hydraulic machines with generators could be found in the literature. Thus, this choice is reasonable, guarantees a good user experience and avoids unnecessary complexity of the code. The code allows for combining as many parallel-arranged units consisting of hydraulic machines, generators, and transformers as required. This feature is essential because it is inherent to the problem to be solved. The option for using a single unit is realised in the code as well, although the solution to this problem is trivial. The simple layout of one unit is shown in Figure [3.1](#).

Power output of the hydraulic machine

After the basic layout has been defined, each component of this layout is described. Three main types of runners for hydraulic machines exist, as mentioned in [4]. The working principles of Francis and Kaplan runners compared to Pelton runners are fundamentally different. Since transient flows are neglected in the present approach, it is possible to avoid looking into the exact working principles of each runner type. It was decided that the basic power Equation 2.2, which is applicable for all runner types, is sufficient for the task. The equation has as parameters the density, ρ , and the gravitational constant g . These two can be adjusted before the optimisation and are assumed to be constant. The variables in the equation 2.2 are the efficiency, η , the water head, H , and the volume flow rate, \dot{Q} . The latter two are scalar inputs to the neural network. The water head and the volume flow rate are given values that describe the current state of the hydropower power plant. The water head has an upper and lower limit defined by the upper and lower reservoirs of the power plant or the tailwater rating curve. However, in this stage of development the head is assumed to be constant. The volume flow rate is limited by the hydraulic machine itself. These limits are defined upfront and cannot be exceeded by the developed computer program. The efficiency is determined by the computer program according to a given efficiency table A.1. Such a table is the foundation for an efficiency hill chart described in Chapter 2.1.2. The computer program is able to interpolate the current efficiency from the efficiency table by receiving the current volume flow rate and water head. Currently, the rotational speed is fixed and cannot be varied. In a future developing step, the rotational speed could be implemented as an additional input.

Power output of the generator

The generator is incorporated as an additional efficiency factor. As previously stated in Section 2.1.2, the generator efficiency can be described as a function of the load factor and power factor. As an approximation, the generator efficiency η_g is provided as a table A.4, which sets the relative power input to the generator in relation to the generator efficiency. The actual value is acquired through interpolation. As input serves the mechanical power output of the hydraulic machine, described in the paragraph above, divided by the nominal power of the hydraulic machine. Alternatively, the generator efficiency can also be provided as a fixed value.

Power output of the transformer

The transformer efficiency curve was explained in Section 2.1.2, which remains nearly constant over its operating range. Similar to the generator, the transformer efficiency is obtained through interpolation from a table, which sets the relative power input in relation to the transformer efficiency A.9. For simplification, the relative power of the generator is directly reused for the transformer. Similar to the generator, the transformer efficiency can also be set to a fixed value instead of interpolation.

Objective function

To find the optimal solution to the problem, the RBFNN needs an objective function which shall be optimised. By combining all the above efficiencies, the final objective function is obtained, which can be written as,

$$\max(P_{tot}) = \max(\rho g H \dot{Q} \eta_h(H, \dot{Q}, \omega) \eta_g(P_m) \eta_t(P_e)). \quad (3.1)$$

The maximum of this function has to be determined for each possible operating point. Section 3.2.1 provides a more detailed description.

3.2 Code structure

The program has two main modes: Optimisation and operation. Optimisation is required for any newly defined hydropower plant. If optimisation was completed before, the saved output can be reused for running the RBFNN with discharge time series and calculating, for example, the yearly power generation.

3.2.1 Optimisation mode

The program structure in optimisation mode is visualised as a flow chart in Figure 3.2. The computer program can be divided into three sections: Data preparation, optimisation, and Evaluation. During Data Preparation, the computer program imports all files, which are provided in a folder structure as ".xlsx" files. The "import data" function imports all files and saves their content in arrays. The

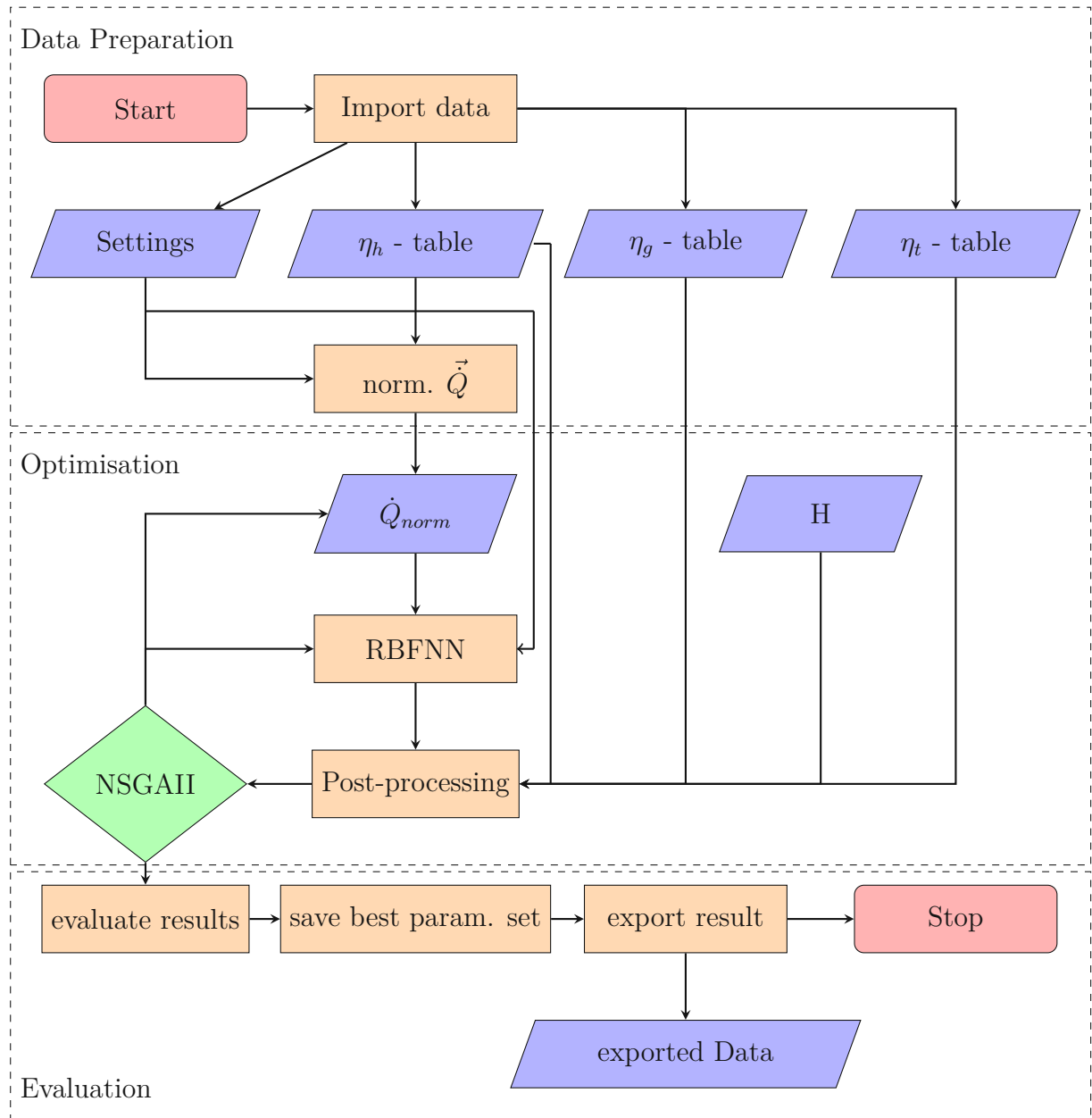


Figure 3.2: Flow chart - optimisation mode

settings data and efficiency table data are used by the "normalisation \vec{Q} " function. It calculates the maximum possible volume flow rate of the total power plant and normalises the vector. Thereby, the volume flow rates always take values between zero and one, which is the required input format for the RBFNN. Optimisation is happening in a loop, which is controlled by the NSGAI algorithm. The optimiser is responsible for generating the best parameters for the RBFNN and deciding when to terminate the optimisation loop. A detailed explanation is found in Section 3.2.2. Noteworthy is that the RBFNN is always receiving a single element of the normalised volume flow rate vector at a time. The post-processing module receives the RBFNN's output as well as the current water head as inputs to evaluate the efficiencies correctly. If the termination criteria are met, the NSGAI algorithm interrupts the optimisation. The result evaluation function receives all parameter sets from the NSGAI algorithm. It decides on the best parameter set depending on the objective function 3.1, which is the summed power output of all discretisation steps. The best parameters set is run once with all discretised volume flow rates from the normalised \vec{Q} and exports this data for later analysis. The set itself is saved and exported as well. Then, the program is stopped.

Program core: Development of the RBFNN model

In the course of this thesis, a neural network was developed suitable for the specific task. The goal is to approximate a function describing the output power of a hydropower plant. Due to the great capability of RBFNNs in function approximation, this type of network was chosen 31. The network went through several development stages. Only the final stage is presented here. The focus was on finding a trade-off between a neural network which is complex enough to fulfil the task while keeping the optimisation time down to a minimum. The final neural network layout, the evaluation of the efficiency factors and the calculation of the power output are sketched in Figure 3.3. Circles visualise network nodes, and rectangles stand for functions and variables. The core of the computer program code can be separated into two modules, the RBFNN itself and a post-processing module. The latter is required because the neural network was not trained as in machine learning with training data, but optimised with an optimiser algorithm.

RBFNN - Module

The neural network is a feed-forward network that consists of three layers and one

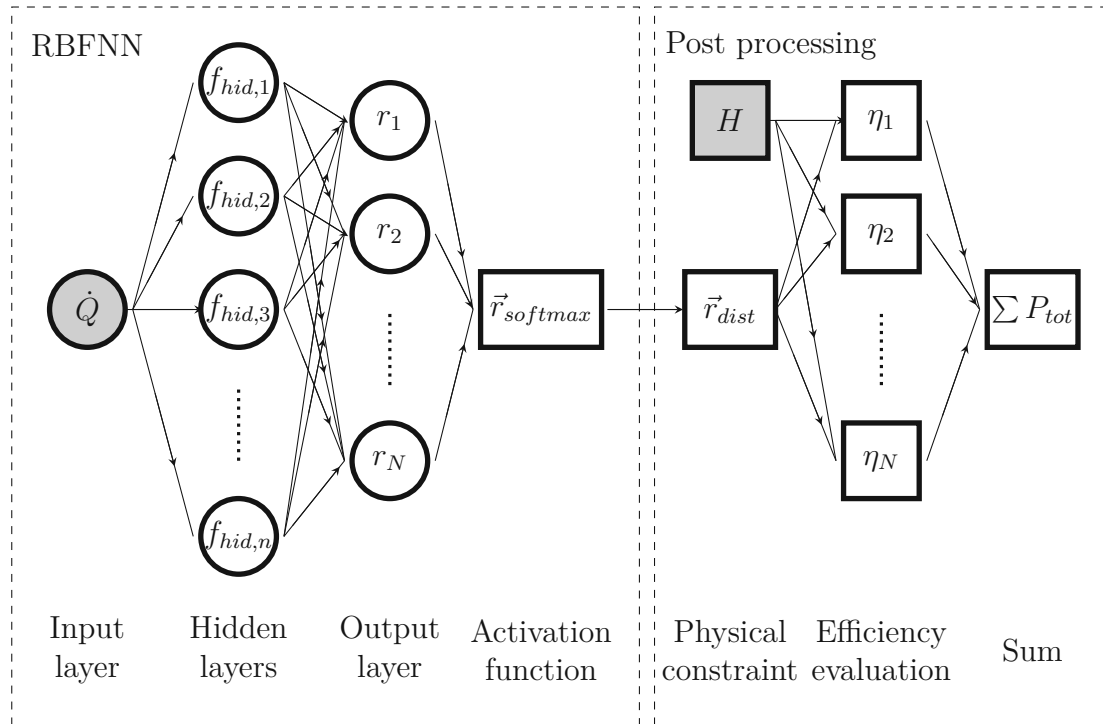


Figure 3.3: Schematic view of code structure

activation function:

- Input layer
- Output layer
- Hidden layer
- Activation function

The Input layer is always a single node which receives the current volume flow rate as the input variable. The current volume flow rate refers to the discretization step of the volume flow for the total power plant, which is transferred to the RBFNN. Since the discretization steps are handed over one after another, "current" refers to whichever step is currently being handed over. The volume flow rate is normalised according to the maximum discharge of the whole hydropower plant, to always be in the interval of $[0; 1]$. The input node has a connection to all hidden nodes. The hidden layer consists of a certain number of hidden nodes, depending on the number of hydraulic machines in the hydropower plant. The hidden nodes accommodate

the radial basis functions, which can be written as,

$$f_{hid,i}(\dot{Q}) = e^{-\frac{(\dot{Q}-\mu)^2}{\theta^2}}, \quad (3.2)$$

where μ is the centre of the specific RBF, θ is the centre of exactly the same RBF, and \dot{Q} is the current flow rate. The index i is defined over the interval $i \in [0, n]$, with n being equal to the number of hydraulic machines N multiplied by the factor of 9, i.e. $n = 9N$. For the application, a Gaussian radial basis function was used. The Gaussian function is favourable as an RBF because it is compact, positive, and converges to zero when its input approaches infinity. Thus, it does not need to be limited on any side. Helpful with the selection of RBF is the fact that the approximation capability of an RBFNN is not dependent on the type of RBF. [31] Each hidden node has a connection to one output node, called the connection weight of i -th node ω_i .

The number of output nodes corresponds to the number of hydraulic machines, N , of the simulated hydropower plant. The reason for this selection is that one output weight, r_j with $j \in [1, N]$, per output node is required, which is representative of the relative amount of optimal water discharge that the corresponding machine shall receive. The output of the output nodes can be written as,

$$r_j(\dot{Q}) = \sum_0^n \omega_i e^{-\frac{(\dot{Q}-\mu)^2}{\theta^2}}. \quad (3.3)$$

A node weight ω is always associated with exactly the same node connection between hidden nodes and output nodes. The rest of the term is the output of a hidden node, equivalent to Equation [3.2]. The output of the RBFNN is, therefore, a vector of output weights, \vec{r} , with the dimension $\dim \vec{r} = \mathbb{R}^N$.

In a standard neural network, each output node has an activation function, as explained before in Section [2.2.1]. However, a special activation function was used for the present task. The neural network receives a normalised volume flow rate as input, which cannot exceed 1. The output of the neural network shall be the vector \vec{r} , whose elements' sum shall always be 1. By multiplying the input by the summed output, the total current volume flow rate is obtained. To achieve this, the Softmax function was used. The function was first used in machine learning by Bridle, who explained its use as follows: "For any input, the outputs must all be

positive and they must sum to unity." [5]. The softmax function is defined as:

$$\vec{r}_{softmax} = \frac{e^{r_j}}{\sum_{k=0}^N e^{r_k}}. \quad (3.4)$$

The softmax function can be described as an exponential normalisation of inputs [5]. Conveniently, the number of inputs is arbitrary. The softmax function receives the RBFNN's output nodes output \vec{r} as input, and provides $\vec{r}_{softmax}$ as output. The function allows for a resizing of the output vector to a magnitude of 1. Because of the exponential function in the softmax function, small elements of the output vector get neglected, smoothening the output vector. This characteristic justifies the name of the function as an activation function. It can deactivate nodes that do not exceed a certain threshold by introducing a variable threshold that depends on the magnitude of all outputs. The variable threshold is what distinguishes it from standard activation functions like the sigmoidal function, mentioned in Section 2.2.1. The softmax function is an activation function combined with a vector normalisation.

Post-processing - Module

The output of the Softmax function $\vec{r}_{softmax}$, contains the distribution for the current water flow discretization step. The optimiser does not include any constraints on the maximum volume flow rates on the individual hydraulic machines. The objective function has a single objective: the maximisation of the power plant's total power output.

Therefore, the algorithm is only limited to distributing the power plants' maximum possible water flow to the hydraulic machines, without being bound to the individual limits of each machine. Because of that, the post-processing module was developed. It is responsible for complying with the physical limits of the hydraulic machines.

If the neural network was trained by training data, the preprocessing module would not be required. The output vector could be directly compared to the training data. Since the training data had to stick to physical constraints, the network would learn to behave the same way. Optimisation with an optimising algorithm requires some additional artificial constraints. The described tasks are handled in the post-processing module by the physical constraint function, as shown in Figure 3.3.

This constraint checks after each run of the RBFNN, if any of the hydraulic machines receives more water than its specified limit. If this is the case, the surplus is calculated and split among not fully utilised units. The algorithm behind this physical constraint function is fairly simple. It does not account for any efficiency gains. However, it acts like a penalty for the optimisation algorithm. The optimiser is hindered from adjusting the RBFNN's weights in a direction that does not fulfil the constraints of the individual units.

The output r_{dist} of the physical constraint function is the final distribution of water flow. It is also a vector with the dimension $dim \vec{r}_{dist} = \mathbb{R}^N$. The second function in the post-processing module is the efficiency evaluation. This function receives the vector from the physical constraint function as one input and the current water head as the second input. The water head, which at this stage stays constant, is necessary to determine the correct efficiency value at the current flow rate. As described in Section 3.1, the efficiency of the hydraulic machine is interpolated from a previously provided efficiency table. One unit consists of the hydraulic machine, the generator, and the transformer, as shown in Figure 3.1. The power output is used for the calculation of the generator efficiency and the transformer efficiency. Those two efficiencies can either be determined by interpolation from efficiency tables or provided as fixed values. The equation

$$P_{obj} = \rho g \sum_i^N H \dot{Q}_i \eta_{h,i} \eta_{g,i} \eta_{t,i} = \rho g \dot{Q}_{plant,max} \sum_i^N H r_{dist,i} \eta_{h,i} \eta_{g,i} \eta_{t,i} \quad (3.5)$$

serves as the objective function for the optimisation algorithm. The total power is the output of the code's core. It is a single scalar, which is returned to the post-processing module and handed over to the optimiser.

3.2.2 Parameter evaluation

Because no training data was available for training the neural network, it was decided to use an optimiser for determining the optimal parameters of the RBFNN. The program is written in Python, leading to the Platybus module, which is a framework for generic algorithms. Platybus focuses on multi-objective algorithms and supports the NSGAII algorithm [16]. The NSGAII optimiser needs only a few inputs. The initial population size was provided, the number of generations as an

additional stopping criterion and the objective function of the problem.

The neural network layout depends on the problem that should be solved. The hidden layer gets equipped with $9N$ nodes, which means the network has $9N$ - RBFs, with N being the number of units in the power plant. This fact is important to note for the correct optimiser selection. One radial basis function has a centre μ , and a width θ , summing up to two parameters per node. Furthermore, a hidden node is connected to all output nodes, and each of these connections is equipped with a connection weight. This leads to a total of $9N * N$ connection weights, which are also parameters of the RBFNN. To sum up, the optimiser receives $2 * 9N + 9N^2 = 9N(2 + N)$ parameters to optimise the RBFNN. These parameters get handed over to the optimiser as decision variables. The objective to be optimised is the total power output of the hydropower plant, as described in Equation [3.5](#). The problem can be described as single-objective multi-variable optimisation.

The NSGAI algorithm is a computationally efficient, generic algorithm with a fast sorting process. Because the number of decision variables is growing exponentially with the number of machines, the optimisation can get computationally intensive for power plants with many hydraulic machines. To allow for optimisation on a standard computer, the NSGAI algorithm offers a great opportunity.

The computer program code structure for the optimiser is shown in Figure [3.4](#). The core of the computer program is now simplified by a dashed box, which houses the RBFNN and the post-processing modules. As inputs serve the volume flow rate and the water head. The output consists of the vector of optimal water distribution, the corresponding total efficiency factor, which consists of η_h, η_g, η_t , and finally, the total power output of the hydropower plant. All parameters of the RBFNN are explicitly displayed in the RBFNN module box. Underneath, the NSGAI is visualised as a separate box. As input, it receives the total power output of the hydropower plant and the current volume flow rate.

To evaluate the parameters in such a way that the neural network is capable of mapping the whole operating range of the hydropower plant, the total range of volume flow rate and water head has to be fed to the optimiser. For that reason, the user has to provide the limits for each hydraulic machine beforehand. The program calculates the total maximum and minimum volume flow rate and creates a discretised vector of volume flow rates. This vector is normalised, and each

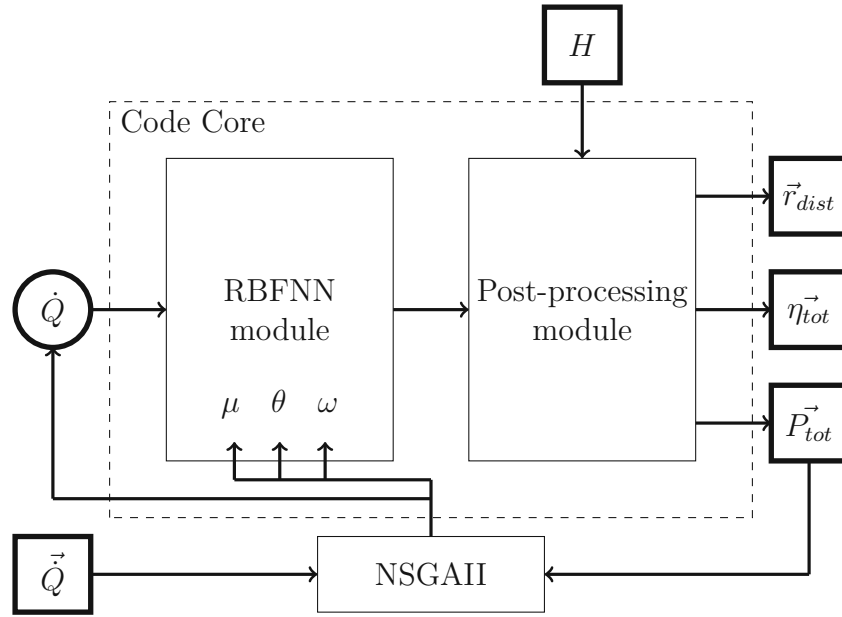


Figure 3.4: Optimiser structure

element is handed over to the optimiser individually. The optimiser then feeds these elements to the computer program core.

3.2.3 Operation mode

The operation mode is currently designed as a single-time run-through application. It is used for calculating the power output for a predefined volume flow rate time series. The operation mode can be divided into two sections: Data Preparation and Operation. During Data Preparation, the program imports all files as before in the optimisation mode. However, this time there are two new imports added: The volume flow rate time series $\vec{Q}_{external}$, and the best parameter set, which was calculated during optimisation. The time series can be provided as an input with volume flow rates arbitrarily discretised over time. The volume flow rate table gets imported and saved in an array. During optimisation, this vector was calculated by the program itself, because it was only a discretisation of the total possible volume flow rates. The volume flow rate time series gets normalised with the help of the power plant limits. The normalised volume flow rates are provided as input to the RBFNN one by one. The RBFNN is initialised with the best parameter set of the optimisation mode. It is a fixed transformation of the input to the optimal relative distribution. The post-processing and export functions work exactly the same as

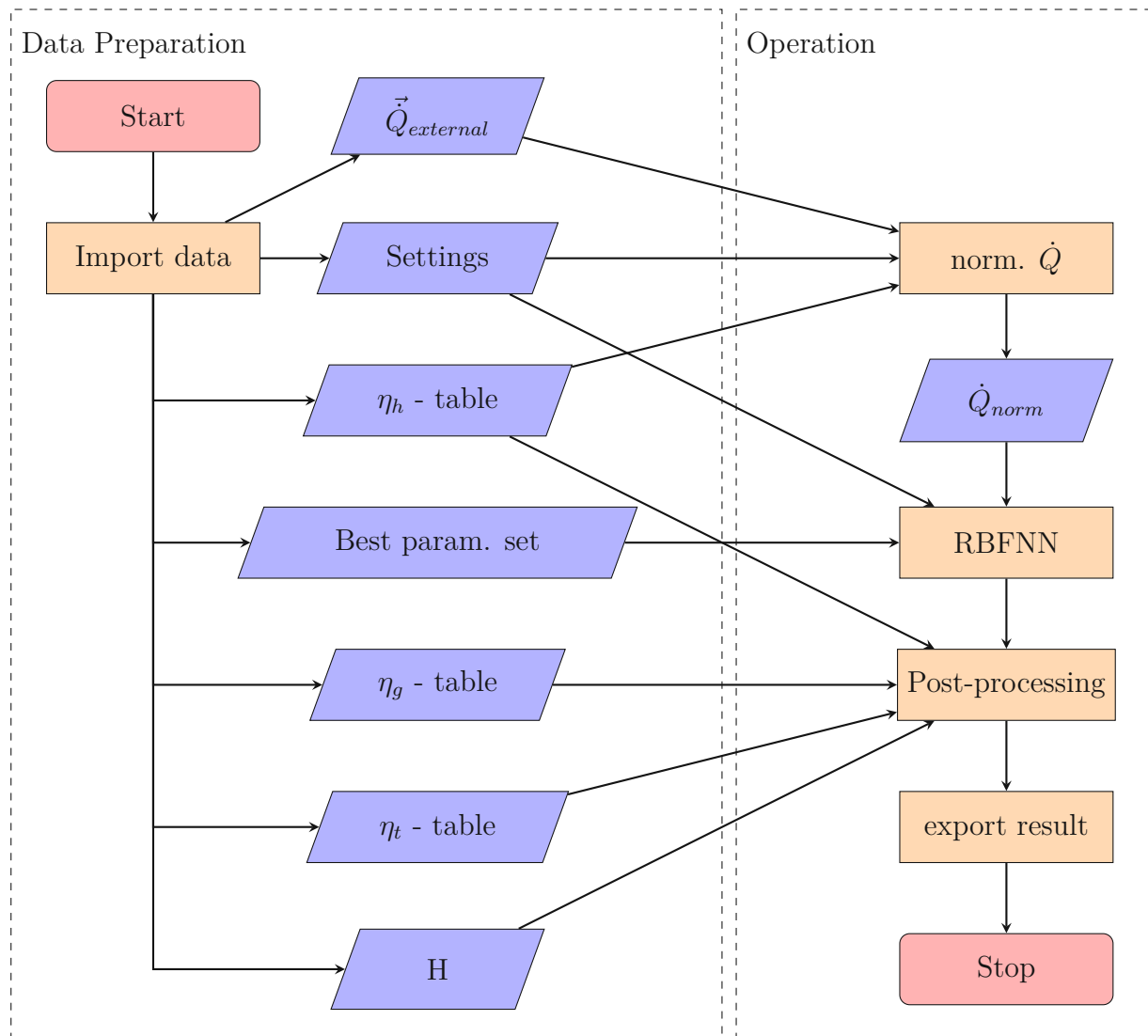


Figure 3.5: Flow chart - Operation mode

before during optimisation. After the total series run through the RBFNN, the program is stopped.

Another possible usage would be a life application of the RBFNN, where a volume flow rate is provided to the network, and it gives back the optimal distribution.

Chapter 4

Results

This chapter presents the optimisation results of the RBFNN. It starts with a simple example of two hydraulic machines and compares it to an optimal solution, computed by brute force. Then, two examples of how the optimal settings for the system were found. Finally, a more complex example is presented, highlighting the limitations of the program.

4.1 Basic example

This chapter describes the first approach for validating the program's output. The program was developed in stages. The first stage only considered the hydraulic machines. It was decided to be the most crucial part of the development, because the hydraulic machine's power output depends on the volume flow rate, squared. Additionally, finding the hydraulic machine's efficiency is challenging. To allow for validation of the results, certain simplifications were made, which are valid for all results presented in Chapter [4.1](#):

- Fixed rotational speed
- Fixed water head
- Same efficiency table for all hydraulic machines
- Penstock efficiency of 100%

- Generator efficiency of 100%
- Transformer efficiency of 100%
- No minimum inflow for the hydraulic machine was set

Results from early development stages were reproduced with the final program by copying the system parameters and inputs. This allows for direct comparison between the results and for ruling out deviations due to different development stages of the code.

The basic example presented in this chapter is a fictitious power plant developed for an AFRY project. It represents a run-of-river power plant. For the project it had to be decided if one hydraulic machine with a maximum inflow of $30 \frac{m^3}{s}$ or two hydraulic machines with a maximum inflow of $15 \frac{m^3}{s}$ each, would generate more power due to a higher efficiency of the hydraulic units. According to the simplification described above, a simplified efficiency table could also be used for the calculation, which is valid for a fixed water head and rotational speed, leaving only the volume flow rate as the input variable. The efficiency values for the machines with a maximum flow rate of $15 \frac{m^3}{s}$ are stated in Figure [A.1](#).

The calculation result is presented in Figure [4.1](#). The solution is simplistic in the way that the RBFNN output r_1 receives 100% of the volume flow until its limit of $15 \frac{m^3}{s}$ is reached. Then the volume flow gets distributed equally on both units, resulting in $r_1 = r_2 = 50\%$.

Three observations merit attention:

1. The transition of r_1 from 1 to 0.5, and of r_2 from 0 to 0.5, seem to describe a step function. However, the RBFNN cannot describe a perfect step function. By examining the data, it becomes apparent that the graph has rounded edges. The impact of this fact is described later on.
2. The plot shows the RBFNN outputs r_i and $r_{dist,i}$. The plot shows that the previously described post-processing of the RBFNN outputs becomes visible (see Chapter [3.2.1](#)). The graphs of r_1 and $r_{dist,1}$ overlap perfectly despite the interval $Q = [15; 16.5]$. Here, the final distribution was adjusted to comply with the maximum volume flow of the hydraulic machine with index 1.
3. There is a slight deviation from the equal distribution of 50% at $17 \frac{m^3}{s}$.

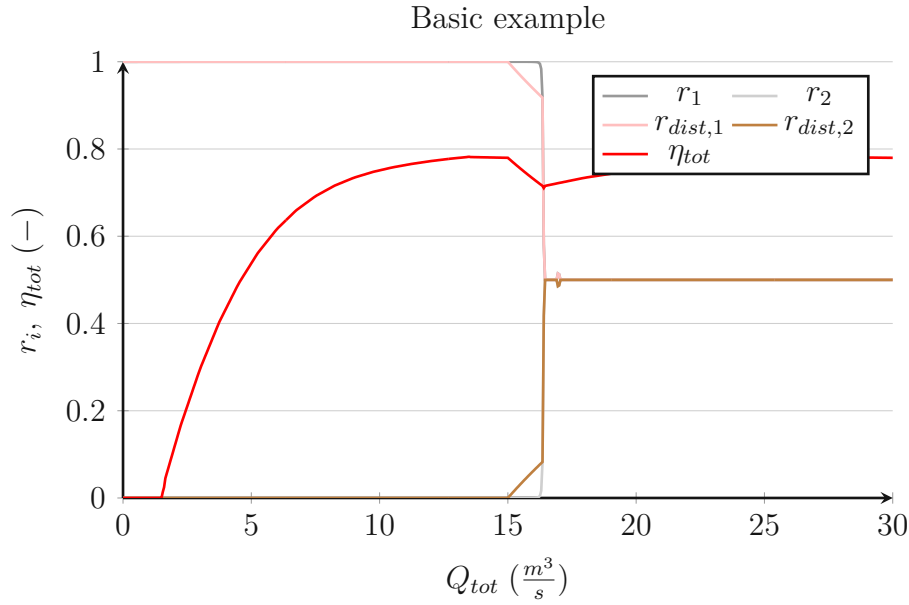


Figure 4.1: Plot of the RBFNN's output r_i , and the total efficiency factor of the hydraulic machines η_{total} over the total volume flow rate Q_{tot}

The red graph of the plot displays [4.1](#) the power plant's total efficiency. The total efficiency factor η_{tot} can be derived from equation [3.5](#) leading to

$$\frac{\sum_0^N \rho g H q_i \eta_{h,i}}{\rho g H Q_{tot}}. \quad (4.1)$$

Because no minimum flow rate was defined, the efficiency curve follows the efficiency table of the hydraulic machine perfectly between $Q = [0; 15] \frac{m^3}{s}$. Afterwards, a section follows which is approximately linear. This section is linear because the volume flow rate of unit 1 is kept constant, as it reached its maximum inflow limit, and the remaining volume flow is not enough for unit two to generate any power output. At a certain point, which will be referred to as *turning point*, the volume flow of both units jumps to an equal distribution. The *turning point* will be explained in more detail in Chapter [4.1.1](#).

The objective function defined in Equation [3.1](#) is plotted in Figure [4.2](#) as a red graph. If point $a \in [0; 30]$ is an arbitrary volume flow, P_{obj} is equal to the accumulated sum of all P_{tot} from 0 to a . Therefore, the relation between P_{tot} and P_{obj} is

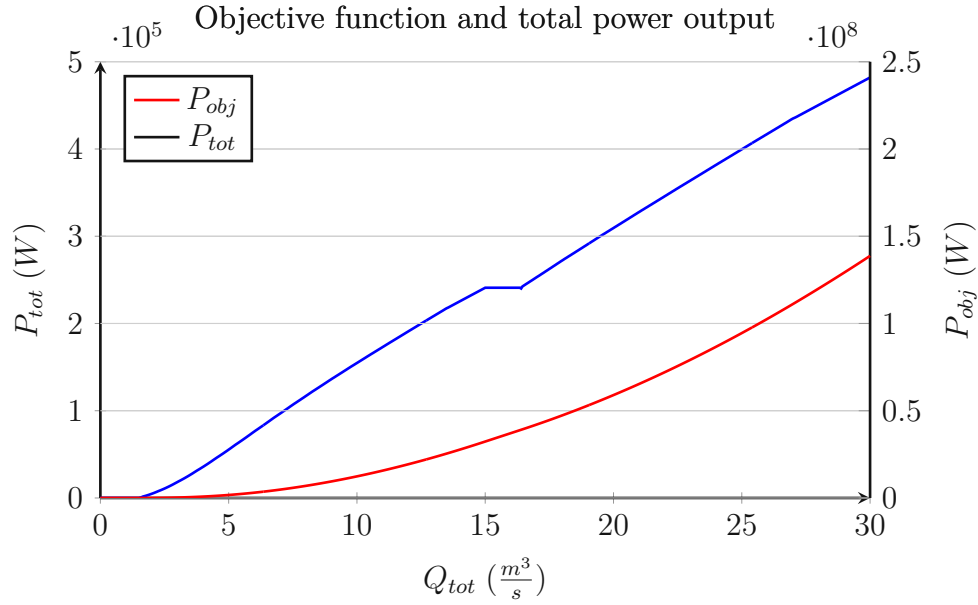


Figure 4.2: Plot of the objective function P_{obj} , and the total power output P_{tot} over the total volume flow rate Q_{tot}

$$P_{obj}(Q_{tot} = a) = \sum_{i=0}^a P_{tot,i} . \quad (4.2)$$

The blue graph in Figure 4.2 shows that the power is increasing nearly continuously with increasing volume flow until the maximum of one unit is reached. The second unit has no power output in the transition between $Q = [15; 16.5]$ because the efficiency table states an efficiency factor of 0. As soon as both units are used with an equal distribution, the power again increases with nearly the same continuous rate. Zooming in on the P_{tot} graph makes a mistake in approximating the optimal solution visible. It is plotted in Figure 4.3.

The power output deviates from the approximate linear function at the discretisation step of $16.4 \frac{m^3}{s}$, marked as a red dot. The reason for this deviation is that the RBFNN's parameters and weights were not optimised enough to fully make the step from an unequal distribution to an equal one in a single discretisation step of $\Delta Q = 0.05 \frac{m^3}{s}$.

In the interval $Q = [16.95; 17] \frac{m^3}{s}$ the distribution of $r_{dist,i}$ deviates from 0.5. The reason for this is explained in Section 4.1.1.

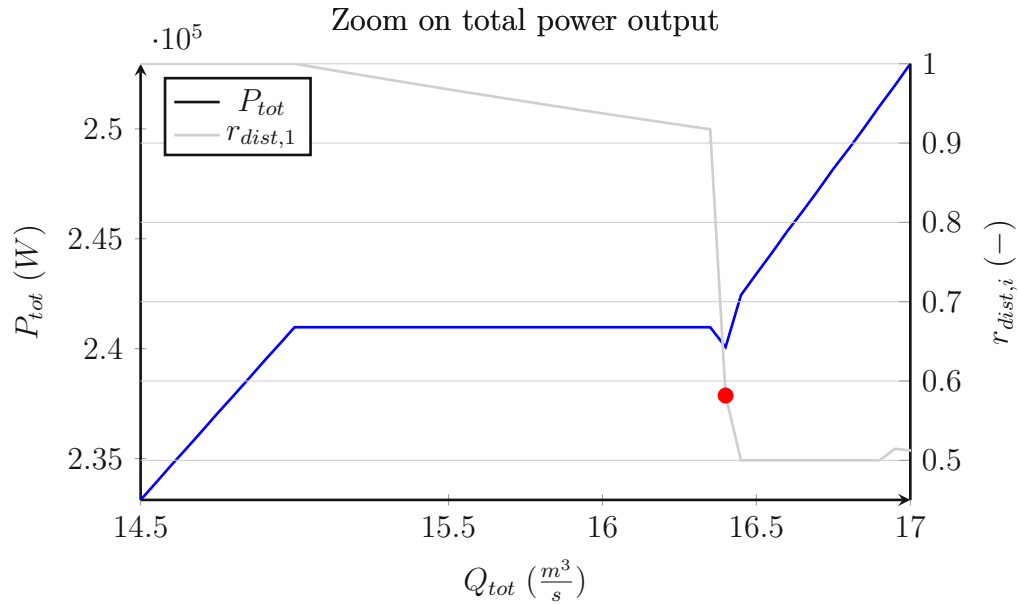


Figure 4.3: Zoomed view on the plot of the total power output and the relative distribution on machine 1, with a red mark at $r_{dist,1}(Q_{tot} = 16.4 \frac{m^3}{s})$.

4.1.1 System Validation

An additional code was developed to validate the results just presented in Chapter 4.1. This code is a relatively simple algorithm, which utilises the same power equation 2.2 and efficiency tables as the basic example. It receives a vector from 0 to 1 in steps of 0.001 as an additional input. This vector is used for discretising the volume flow Q_{tot} into 1000 possible distributions like $r_{dist,1}$ and $r_{dist,2}$. All power outputs are calculated with these distributions, and the maximal one is selected. That way, the highest power output is found. Therefore, a search algorithm calculates the maximum possible power output by brute force. This approach is fundamentally different to the RBFNN. The validation method optimises each discretisation point individually without considering the previous or preceding ones. The RBFNN optimises a continuous function to gain the maximal power output over the whole input range.

In Plot 4.4, the optimal distribution weights $r_{man,1}$, $r_{man,2}$ of the validation method and the corresponding total efficiency $\eta_{man,tot}$ are plotted over the volume flow rate Q_{tot} .

For $Q_{tot} = [0; 16.3] \frac{m^3}{s}$, the results of both methods are nearly identical and confirm

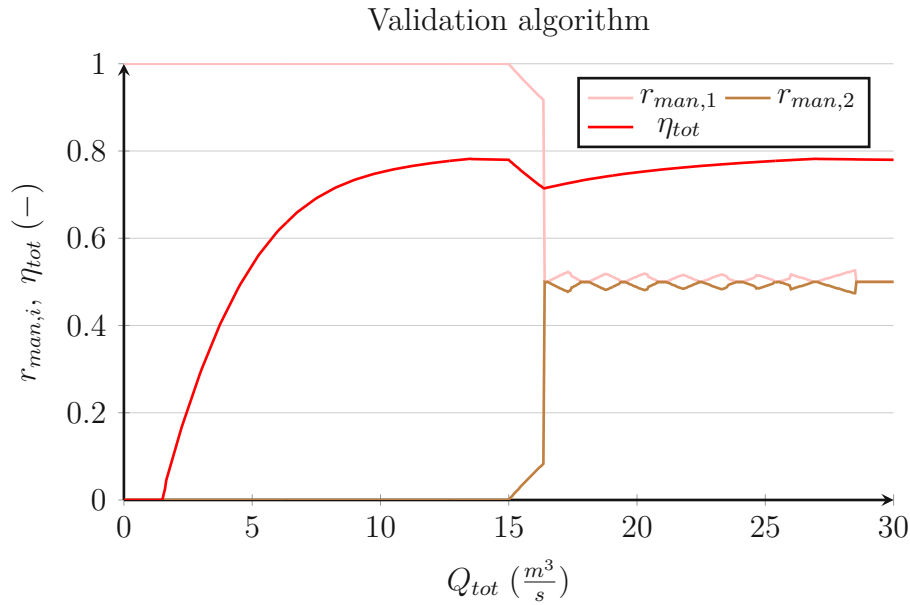


Figure 4.4: Plot of the validation algorithm's output $r_{man,i}$, and the total efficiency factor of the hydraulic machines η_{total} over the total volume flow rate Q_{tot}

each other. Even the transition from one active unit to two units is calculated sufficiently by the RBFNN. As visualised in Figure 4.5, the power curves do not line up perfectly at $Q_{tot} = 16.4 \frac{m^3}{s}$, as foreseen in the previous section. For $Q_{tot} = [16.5; 30] \frac{m^3}{s}$ the graphs seem to line up perfectly again. By summing up all power outputs over the whole volume flow rate range, which corresponds to the objective function of the RBFNN 3.1, the results can be compared regarding their performance. The validation method achieves a power output of 138,739,932.5W, while the RBFNN only manages to achieve 138,738,854.9W. The RBFNN therefore achieves 99% of the power of the validation method's result.

The results are also confirmed by comparing the actual distributed volume flows q_i . In Figure 4.6, the normalized volume flows $q_{norm,i} = \frac{q_i}{Q_{tot}}$ of the RBFNN and $q_{man,norm,i} = \frac{q_{man,i}}{Q_{tot}}$ of the validation algorithm are presented. The plot shows a linear increase in the volume flow rate on the first hydraulic machine, followed by a constant section in combination with a linear increase in the volume flow rate on the second hydraulic machine. In the interval $Q_{tot} = [16.5; 30] \frac{m^3}{s}$ follows a linear increase with half the inclination of the first linear function.

Eye-catching is the graph course of the validation algorithm, which might already

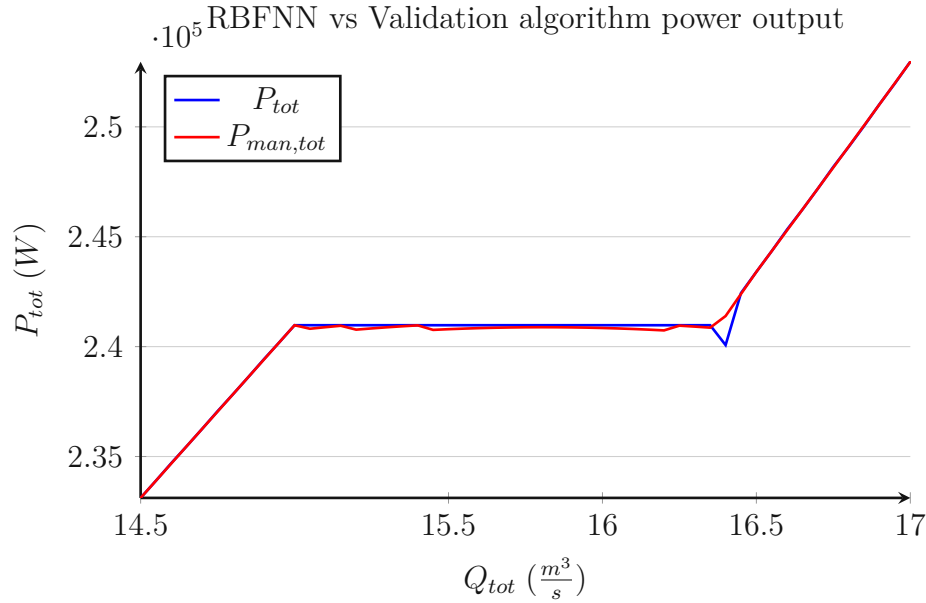


Figure 4.5: Comparison of the RBFNN's and validation algorithm's power output in the interval $Q_{tot} = [14.5; 17] \frac{m^3}{s}$

have been noticed in Figure 4.4. The weights $r_{man,i}$ and thus also the volume flow rates $q_{man,i}$ are not constant in the range of $Q_{tot} = [16.5; 30] \frac{m^3}{s}$. The reason for these fluctuations is of a numerical nature, which becomes visible by plotting not just the optimal solution of the validation algorithm, but all of them.

Plot 4.7 shows a selection of power outputs at fixed total volume flow rates over all possible $r_{man,1}$. With these visualisations, it is possible to explain the just described fluctuations of $r_{man,i}$, as well as the mistake of the RBFNN at $Q_{tot} = 16.4 \frac{m^3}{s}$. All previous plots, where Q_{tot} was on the abscissa, showed three phases: Distribution on a single unit, transition phase, and distribution on two units. These phases are also visible in Figure 4.7. The first phase, where the total volume flow is distributed on a single unit, is outstanding due to its global maxima at $r_{man,1} = 0$ and $r_{man,1} = 1$. During this phase, a small local maximum is noticeable at $r_{man,1} = 0.5$. It gets more dominant with increasing power. In the transition phase, horizontal lines appear for the first time. Those mark the limited maximum flow rate through a single hydraulic machine. In this phase, the absolute difference between the local and global maxima decreases because the restricted volume flow rate limits the global maxima. At $Q_{tot} = 16.4 \frac{m^3}{s}$ the global maximum is for the first time at $r_{man,1} = 0.5$.

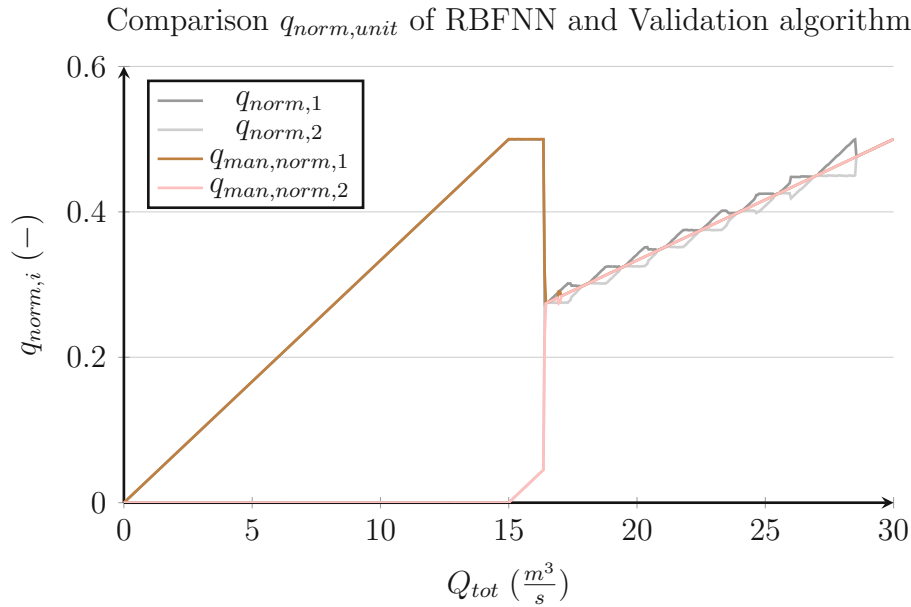


Figure 4.6: Comparison of the optimal distributed, normalised volume flow rates of each unit according to RBFNN $q_{norm,i}$ and to the validation algorithm $q_{man,norm,i}$ over the total volume flow rate Q_{tot}

Because the RBFNN is creating a continuous function from RBFs, this function has to perfectly follow the switch from two global maxima to one global maximum. If the parameters and weights are not trained enough for approximating this switch, the solution will have a slight mistake, as presented in Figure 4.3.

Going back to the fluctuations of $r_{man,i}$, which appear in the third phase. In this phase, the power curves over $r_{man,1}$ have one global maximum, which should be a trivial optimisation task for the algorithm. However, by zooming in on these power curves at $Q = [18, 18.75, 19.5] \frac{m^3}{s}$ it becomes apparent that there is no perfect continuous function, but consists of points which were interpolated. In Figure 4.8, all three curves show edges, which are located where the product of $r_{man,1}$ and Q_{tot} results in a value that is explicitly mentioned in the efficiency table of the hydraulic machines. The efficiency data had to be linearly interpolated between the two edges from the resulting values of $q_{man,i}$. This interpolation includes a small error, which is largest in the middle of the two original points. For this reason, the power output dips in between. For $Q_{tot} = 18 \frac{m^3}{s}$ and $Q_{tot} = 19.5 \frac{m^3}{s}$, the equal distribution falls perfectly on values in the efficiency table. For all values in-between unfortunate not, resulting in slightly better power output for distributions deviating from $r_{man,1} = 0.5$. For $Q_{tot} = 18.75 \frac{m^3}{s}$, this results in a relative power decrease of

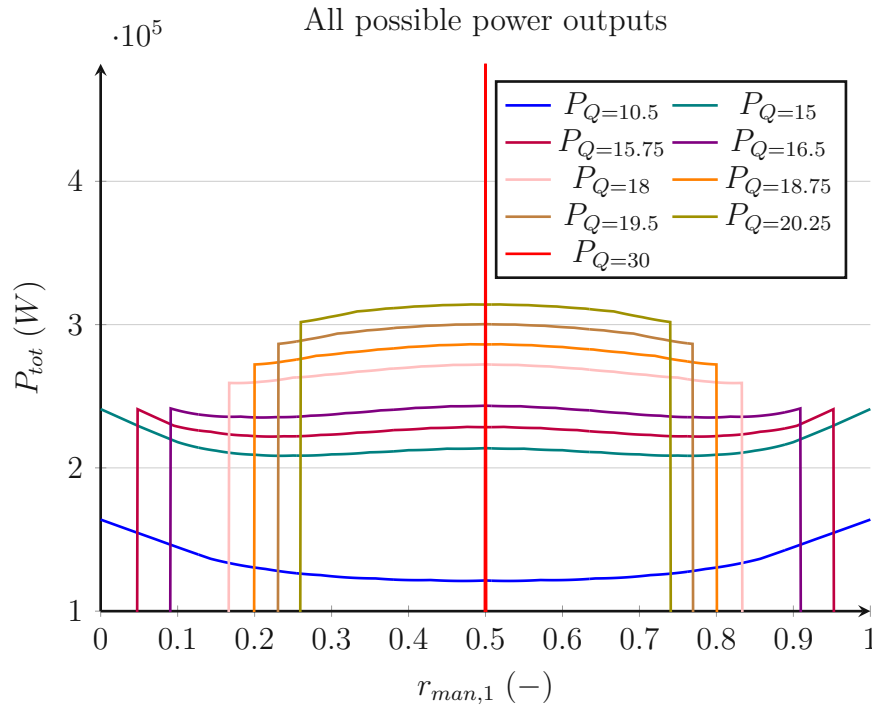


Figure 4.7: All possible power outputs at selected fixed total volume flow rates (shown in the legend) plotted over all possible distributions of $r_{man,1}$.

0,03%, which is seen as negligible for the program's application.

4.2 Setting system parameters

As the system description in Chapter 3 presented, the program consists of three fields: Hydro power, RBFNNs, and NSGAI optimiser. For the latter two, the correct layout, system parameters, and settings had to be researched and evaluated using an empirical study. This work identified two crucial settings, which shall be described in the following two chapters.

4.2.1 Optimal number of nodes

In this chapter, the term node refers only to hidden nodes of the RBFNN. For a review of the structure of an RBFNN, refer to Chapter 2.2.2. The number of hidden nodes of an RBFN can be compared to the number of parameters in regression

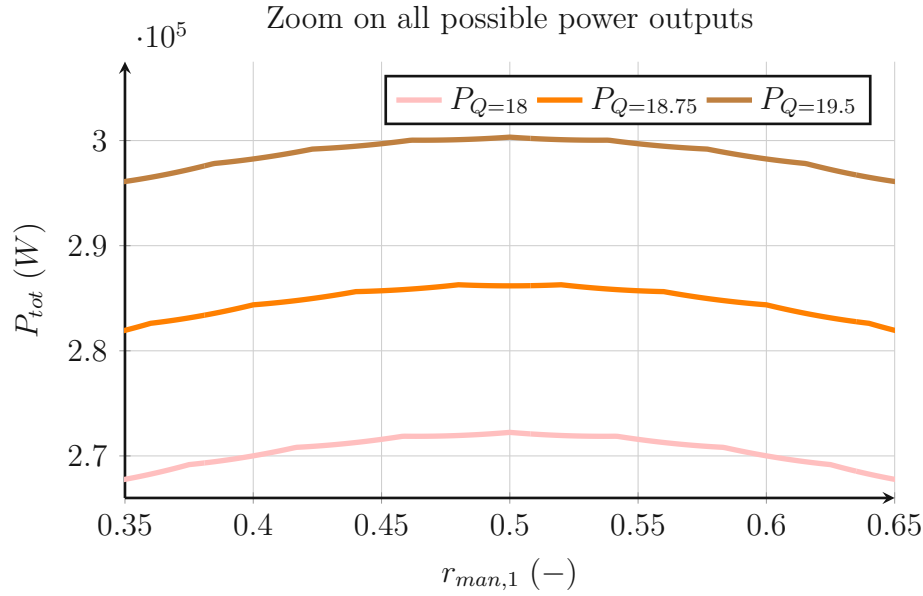


Figure 4.8: Zoomed view on all possible power outputs at $P_{tot}(Q = [18, 18.75, 19.5])$ from Figure 4.7 in the interval $r_{man,1} = [0.35; 0.65]$

models. The number of hidden nodes can be adjusted to match the degree of the function to be approximated [6]. However, as presented in the previous chapter, the objective function can be structured into three parts, and the task for the RBFNN was to approximate a single continuous function. Thus, the degree of the function could only be estimated.

An empirical approach was chosen to identify the required number of nodes for the problem. Choosing too few hidden nodes has noticeable implications on the result. For the example in Plot 4.9, a node number of $3N = 6$ instead of $9N = 18$ was chosen. For enabling a direct comparison, all other parameters were set identically to the basic example from Section 4.1.

The limited number of nodes does not influence the sections of linear increase at low volume flows $Q_{tot} = [0; 15] \frac{m^3}{s}$ and high volume flows $Q_{tot} = [23; 30] \frac{m^3}{s}$. In the transition, the most crucial part of the calculation, the absence of the required number of nodes becomes apparent. The prior described step, from $r_1 = 1$ to $r_1 = 0.5$, shows two problems.

Firstly, the step is not approximated sharply enough to avoid a drop in efficiency and the power output. In the red graph of η_{tot} , the drop is clearly visible.

Secondly, fluctuations in the RBFNN output, r_1 and r_2 , can be observed after

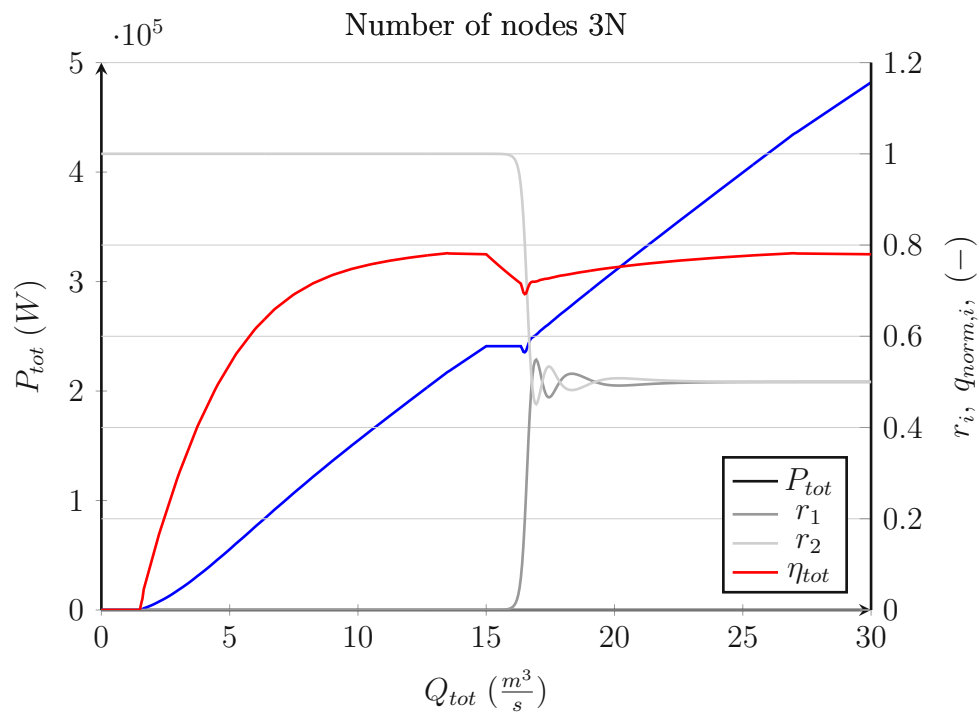


Figure 4.9: Example for using not enough hidden nodes for the function approximation

$Q_{tot} = 16.4 \frac{m^3}{s}$. Because these fluctuations cannot be eliminated through longer optimisation by increasing the number of generations of the NSGAI algorithm, it is assumed that the parameters of the limited number of nodes cannot be set to approximate the optimal solution.

Choosing too many nodes is also possible, although this case is more complicated to detect than choosing not enough nodes. The function approximation will work perfectly fine and give good results. However, the optimisation will take more time than necessary. To find out if too many nodes were used, the RBFs have to be examined directly. As [31] states, too many nodes result in RBFs with a thin width θ and small weights ω . In case of the Gaussian RBF, there will be many thin bell curves, with negligible ω weights.

Several optimisations with only a few nodes were performed to evaluate a good number of nodes. The number of nodes was increased successively until reasonable results were found. This approach is similar to the constructive approach described in [31], which is an automated way to find the correct number of nodes. Afterwards, the RBF's parameters were examined to confirm the results.

Because the degree of the objective function increases with the number of hydraulic machines, an equation was derived that sets the number of nodes in relation to the number of machines. The number of nodes n is recommended to be equal to $9N$.

4.2.2 Finding optimal settings for NSGAI

The second crucial parameter is the setting of the NSGAI algorithm. As described in Chapter 2.3, the NSGAI is a generic algorithm. It has two main input settings: the original population size and the number of generations. During the first runs of optimisations, the population size for the NSGAI was set far too low, resulting in insufficient optimised parameters for the RBFNN. These results were reproduced and are plotted in Figure 4.10.

On first sight, the plot looks similar to the results of Chapter 4.2.1, where insufficient hidden nodes for the RBFNN were used. It makes these two problems difficult to distinguish from each other. For $Q_{tot} = [0; 15] \frac{m^3}{s}$ the RBFNN was optimized perfectly. In the transition phase, the insufficient optimisation becomes apparent. First, there is a drop of r_2 . Followed by a too steep curve, instead of a sudden step of the RBFNN outputs r_1 and r_2 . The result is a drop in efficiency and power

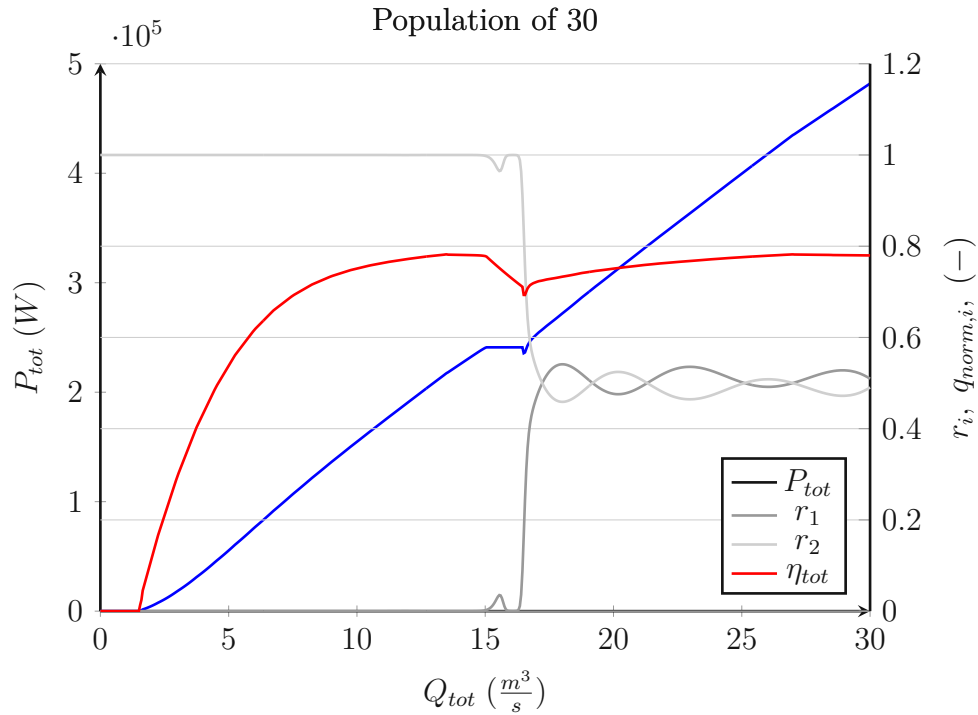


Figure 4.10: Example for using a too small population size for the NSGAI algorithm

output. For $Q_{tot} = [15; 30] \frac{m^3}{s}$, the RBFNN's outputs are fluctuating around their desired value of 0.5. It is the most noticeable difference compared to the case of using not enough hidden nodes, where the fluctuations only occurred for a shorter interval of $Q_{tot} = [16.4; 23] \frac{m^3}{s}$.

Research revealed that the number of nodes and the optimal population size of the NSAGII algorithm are related. As the paper [10] states, the population size should be about four times the number of decision variables in the problem being solved. The optimal population size for any problem in this program can be calculated with

$$Pop_size = 4[9N(N + 2)] , \quad (4.3)$$

where the term in the squared brackets is the number of decision variables, as described in Chapter 3.2.2. While an arbitrarily high population size can be set, it will have an adverse effect on the calculation time. This equation provides a simple estimation.

4.3 Four unit example

A fictitious but more complex example than the basic example was created to test the limitations of the RBFNN. It consists of 4 hydraulic units, each connected to one generator and one transformer. As basis for the efficiency data of the hydraulic machines serves the model tests of an old project executed by AFRY. The generator and transformer data are taken from a comparable project, because no data from the same project was available for all components.

In Chapter 4.1, it was shown that the RBFNN is capable of calculating the optimal solution for two machines with the same efficiency curves. Further simulations showed that more than two units can also achieve the result. However, to add more complexity to the problem, the data was altered between the units. Units 1 and 4 use the original efficiency table for the hydraulic machine, the generator, and the transformer. The efficiency table of Unit 2 and Unit 3 for the hydraulic machines was altered. The generator and transformer efficiency tables are equal to those of Unit 1 and Unit 4.

This experiment aimed to evaluate the program's capability to optimise the RBFNN for a specific application. The investigated scenario involves two hydraulic machines, distinguished by distinct optimal operating points: one machine performs better at lower volume flow rates, while the other achieves optimal performance at higher flow rates. These boundary conditions result in a solution in which, at lower flow rates, the entire volume flow is distributed to the machine with the lower optimal flow rate. Once this machine reached its optimal operating point, there is a certain point at which the entire flow rate gets redirected to the machine with the higher optimal flow rate. To examine this scenario, all efficiency points of Unit 2's efficiency table were altered to occur one discretisation step before those of Units 1 and 3.

The efficiency table of Unit 3 is similar to that of Units 1 and 4, but with an efficiency decrease by a factor of 0.04 over all discrete points. Therefore, this machine should receive water only if all other machines are already used.

In Chapter 4.1, a validation method was introduced. The same approach is also used for this example. Of course, the generator and transformer efficiencies were also implemented in the validation algorithm to guarantee a comparable result. The plots presented below show more abrupt changes compared to the ones in Chapter 4.1 and 4.2, because the discretisation of the volume flow rate was reduced due to

the higher capacity of the hydraulic machines. It allows for shorter optimisation times.

Figure 4.11 shows the optimal solution according to the validation algorithm. All four units have the same maximal volume flow rate capacity of $190 \frac{m^3}{s}$. The plot can be separated into four sections:

1. $Q = [0; 190] \frac{m^3}{s}$
2. $Q = [190; 380] \frac{m^3}{s}$
3. $Q = [380; 570] \frac{m^3}{s}$
4. $Q = [570; 760] \frac{m^3}{s}$

The local minima can easily detect individual sections in the total efficiency η_{tot} curve. The first section highlights perfectly the previously explained motivation for this example's setup. The total volume flow rate is directed to Unit 2, because its optimal operating point is at lower volume flow rates compared to Units 1 and 4. The best efficiency point is an instantaneous switch from Unit 2 to Unit 4. The switch is so abrupt because the efficiency table of Unit 2 was altered by just moving the efficiency values by one discretisation step. In the second section, Unit 2 is used at lower flow rates before Units 1 and 3 receive equal amounts of volume flow rate. In the third section, Units 1, 2, and 4 are used because Unit 3 has the worst efficiency of all units. Further, it is noticeable that the optimal operating point of Unit 2 is lower compared to Units 1 and 4.

In Figure 4.12, the RBFNN's result is plotted. The basic order of the units is the same as that of the validation algorithm. However, it is visible that the network's output uses one machine per section and does not redirect the volume flow as soon as another machine would be more efficient. Further, section three of the four has an equal distribution between Units 1, 2, and 4. Therefore, Unit 2 is not run at its optimal operating point. Finally, by summing up all power outputs of the total range $Q = [0; 760] \frac{m^3}{s}$, which corresponds to the result of the objective function, the two results can be quantitatively compared. The RBFNN achieves a total output of $3.0832 * 10^7 W$ and the validation algorithm $3.1495 * 10^7 kW$. The RBFNN achieves a summed power output that is therefore 2.1% smaller compared to the validation method.

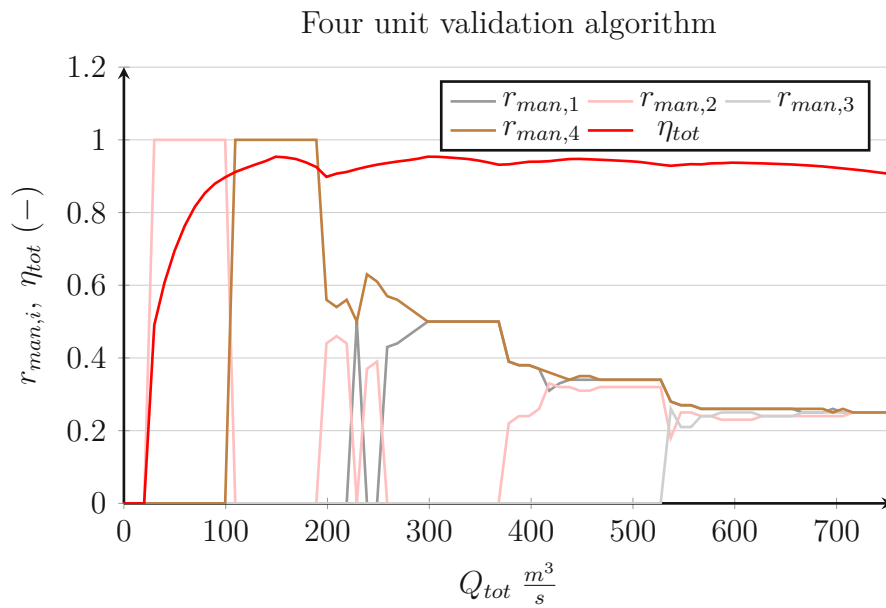


Figure 4.11: Result of validation algorithm for distributing the volume flow over four differing power units, with the resulting total efficiency

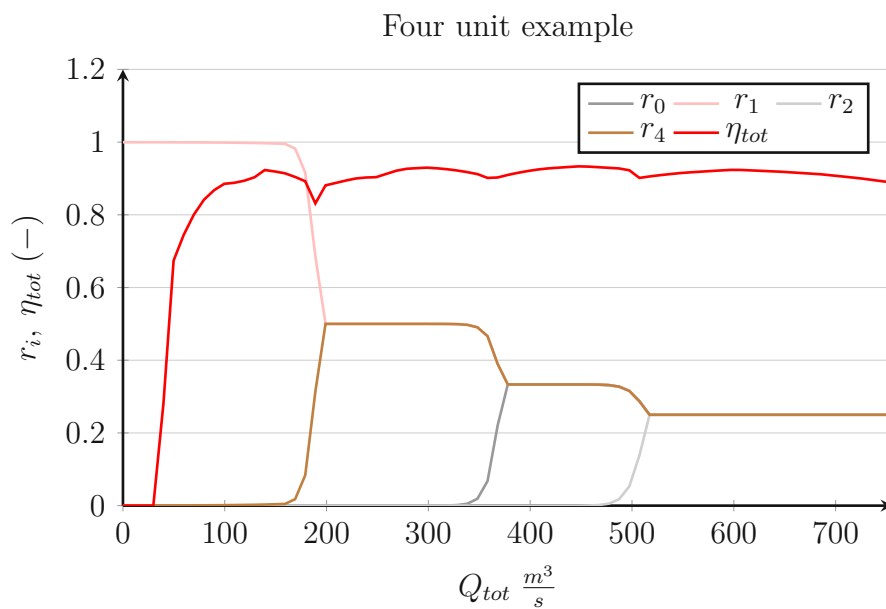


Figure 4.12: Result of RBFNN for distributing the volume flow over four differing power units, with the resulting total efficiency

Chapter 5

Discussion

The results presented show that the RBFNN is able to model the objective function for maximising the power output of a hydro power plant with an arbitrary number of units. The network type was selected appropriately as suggested by the literature. RBFNNs were developed and therefore are well suited for function approximation tasks [17].

For fulfilling the defined task, the network's structure had to be adjusted and its output had to be post-processed. The network structure was complemented with the Softmax function as an activation function. It allowed for an elegant mathematical formulation to resize the network's output vector to a magnitude of 1 while avoiding losing the relative differences of the output elements' magnitudes. The Softmax function, therefore, limited the output of the network to the maximum physically possible volume flow of the total power plant. The post-processing of the network's outputs was necessary because neither the objective function nor the network's structure implied physical limitations of the individual units. The network was optimised with the NSGAI algorithm, which had the maximisation of the objective function as the optimisation goal. Therefore, additional constraints were implemented in the post-processing module.

The results suggest that the network structure has a major impact on the approximation capabilities of the RBFNN. A sufficient number of nodes is required for achieving satisfactory results. Large network structures result in high computational efforts. An equation has been developed for estimating a reasonable network structure, and is presented in Chapter 4.2.1.

5.1 Validation of the RBFNN results

The RBFNN was not able to evaluate the optimal solution as precisely as the validation method used. This was first demonstrated using the basic example in Chapter [4.1](#), where a slight decrease in power output was observed for one discretisation step of the volume flow rate. The reason for the result deviations between the RBFNN and the validation method is most likely the fundamental difference in the output properties of both methods. The RBFNN is able to provide a continuous function. It reaches that goal by optimising the objective function to gain a maximal power output over its whole input range. The validation method just takes discrete steps and evaluates the optimal solution for each one of them. Each discretisation step is evaluated independently of the previous and the proceeding one. Because both results are plotted from discrete points with lines as connections, the visualisation can lead to wrong assumptions that both results have the same properties.

The RBFNN's performance on a more complex example with four hydraulic machines and varying efficiency curves was insufficient. It directed the volume flow rate to the best-performing unit, depending on which had the highest efficiency over its whole range of volume flow rates. However, the RBFNN failed to redirect the volume flow for differing optimal operating points between hydraulic machines as shown in Chapter [4.3](#). Reasons for this behaviour could not be proven during this thesis. Some ideas for potential causes are discussed here.

The RBFNN's structure was found to be sufficient for the task. The RBFNN should be capable of approximating the optimal solution as calculated by the validation algorithm. The observation that the networks' output during various test runs always produced solutions with an equal distribution of volume flow rates across all machines, regardless of their differing optimal operating points, indicates that the underlying problem is the NSGAI algorithm. Two potential sources were identified:

1. The optimiser receives as an objective function the sum of power outputs of the whole input space, but it has to optimise a large number of parameters and weights. These preconditions may make it extremely difficult to converge to an optimal solution.
2. The post-processing module simply cuts off certain areas of the solution space.

It is unclear how this practice influences the optimiser.

Overall, it must be pointed out that the program is inefficient. The computational effort required to optimise a relatively large neural network for this application proves to be high. As the number of units in the power plant increases, the number of decision variables grows exponentially, due to the suggested equation for node number selection. This characteristic leads to scalability issues.

This thesis examined stationary operating points. Taking this into account, it is reasonable to look at each operating point individually to gain the maximum efficiency at this exact operating point. This argument benefits the validation methods approach. The initially assumed advantage of the RBFNN's continuous output turned out to be a drawback in the context of this specific use case.

However, hydro power plants are not operated in the same state continuously. Optimising for just a discrete point can lead to unrealistic operating procedures for real-life applications. Unfortunately, it cannot be proven that the RBFNN result would lead to better outcomes in real applications compared to the validation method.

A benefit of the RBFNN is that it does not optimise for the numerical patterns in the data. No exact reason can be provided for this behaviour. A potential explanation is again the optimisation of the continuous function over the whole input range.

The primary drawback of the computer program is its lack of transparency regarding its solution process. Once the problem reached a certain complexity, it is nearly impossible to trace back how the program came to its result. Due to this limited transparency in combination with its suboptimal optimisation results, it is found to be inappropriate for the intended application.

5.2 Future works

This work concludes that the applied method is inefficient for the given task. However, this finding is limited to the combination of optimising an RBFNN using the NSGAI algorithm for a certain application. Nonetheless, the work lays the foundation for a potential machine learning application for the same problem.

As this work showed, it is possible to optimise the RBFNN to represent a continuous function, which allows for deriving the optimal volume flow distribution on an arbitrary number of hydraulic machines. The simplifications of a fixed water head and fixed rotational speed were introduced to reduce the scope of this work. In a further development step, the introduction of a variable water head and variable speed is feasible. Because the efficiency hill chart changes depending on the water head and the rotational speed, the RBFNN structure might need further adjustment to solve such a problem sufficiently. The verification of this claim is a potential future research question.

The approach to optimise the parameters of an RBFNN with the NSGAI algorithm is inefficient, because the optimiser could also be used directly on the problem. The approach was picked because of the initial motivation to use machine learning. Due to the lack of data, the machine learning approach was abandoned. This work showed that, in principle, the underlying problem could be solved by using an artificial neural network of the RBF type. The structure of the network, as well as its size, can be reused for a machine learning application. The optimiser needs to be replaced by a learning algorithm. Sufficient training data would need to be acquired. Obtaining the required measurements should be straightforward. A potential hindrance could be gaining data with a high enough resolution of all possible stationary operating points. If only data from the usual operation is gained, the machine learning algorithm is likely to just replicate these operating points.

Chapter 6

Conclusion

The aim of the current thesis was to use an artificial network for the optimisation of hydroelectric power generation. The central research question was:

Can an artificial neural network be used to simulate the optimum load distribution in a hydropower plant with an arbitrary number of machine sets?

An appropriate artificial neural network of the Radial Basis Function type was selected as the literature suggested. This type stands out because of its universal approximation capabilities. Due to the lack of sufficient training data, a machine learning approach was ruled out and replaced with an optimisation task. As a simple to use multi-objective optimiser, the NSGAII algorithm was selected. A computer code was developed which consists of a self-developed Radial Basis Function Neural Network, the NSGAII module, and some pre- and post-processing automations.

The network was expected to easily model the optimised objective function of the problem. Then, more complexity could be added by a variable water and a variable rotational speed. By using an RBFNN with a sufficient number of hidden nodes, it was expected to model and optimise any described hydro power plant.

In the absence of validation data, another method for validating the results was developed. A specially developed algorithm calculated the optimal solution to the problem by brute force. The results of both methods were compared to each other.

Although the RBFNN showed potential to model the objective function properly, the final goal was not achieved as expected. By using the validation algorithm,

it could be demonstrated that the computer program is able to approximate the objective function of the problem, but with limited precision. The conclusion drawn is that an RBFNN is capable of approximating a function which defines the optimum load distribution in a hydro power plant with an arbitrary number of machine sets. However, it is questioned if the approach might be inappropriate because of the RBFNN's output type, which is a continuous function. For the application also a discrete solution in combination with interpolation might be sufficient. Additionally, the NSGAI algorithm was found to be inefficient for the application. The computational effort is very high for a system which offers limited to no transparency on its solution process.

The approach showed that RBFNNs are able to fulfil the task. It was not determined if the system could also work for varying head and variable speed applications. Therefore, two fields of future work were identified: The examination of optimising variable water head and variable speed systems, and applying machine learning to the same optimisation tasks using RBFNNs. For the latter, the RBFNN structure developed in this thesis could be reused.

Bibliography

- [1] Acosta, F. M. A. (1995). Radial basis function and related models: An overview. *Signal Processing* 45, 37–58.
- [2] Association, I. H. Types of hydropower, Web Page.
- [3] Blöschl, C. Einfluss der Integration von verteilten PV-Anlagen auf die Alterung von Verteiltransformatoren, Thesis, Technische Universität Wien, 2024.
- [4] Breeze, P., Chapter 8 - Hydropower In *Power Generation Technologies (Third Edition)*, Breeze, P., Ed.; Newnes: 2019, pp 173–201.
- [5] Bridle, J. (1989). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems* 2.
- [6] Buhmann, M. D. (2000). Radial basis functions. *Acta Numerica* 9, 1–38.
- [7] Chenchireddy, K. (2022). A Review on Different Power Transformers. *International Journal of Advanced Research and Innovative Ideas in Education* 6, 2022.
- [8] Chien-Cheng, L., Pau-Choo, C., Jea-Rong, T., and Chein, I. C. (1999). Robust radial basis function neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 29, 674–685.
- [9] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. In *Parallel Problem Solving from Nature PPSN VI*, ed. by Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H.-P., Springer Berlin Heidelberg: 2000, pp 849–858.
- [10] Doerr, B., and Qu, Z. (2023). From Understanding the Population Dynamics of the NSGA-II to the First Proven Lower Bounds. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 12408–12416.

- [11] Dougherty, J., Kohavi, R., and Sahami, M., Supervised and Unsupervised Discretization of Continuous Features In *Machine Learning Proceedings 1995*, Frieditis, A., and Russell, S., Eds.; Morgan Kaufmann: San Francisco (CA), 1995, pp 194–202.
- [12] Energy, U. D. o. What is Hydropower?, Web Page.
- [13] Grigsby, L. L., *The electric power engineering handbook*; CRC Press Boca Raton, FL, USA: 2001; Vol. 2.
- [14] Guo, P., Wang, Z., Sun, L., and Luo, X. (2017). Characteristic analysis of the efficiency hill chart of Francis turbine for different water heads. *Advances in Mechanical Engineering* 9, 168781401769007.
- [15] Gutowski, M. W. (2005). Biology, physics, small worlds and genetic algorithms. *Leading edge computer science research*, 165–218.
- [16] Hadka, D. Platypus - Multiobjective Optimization in Python, Web Page, 2024.
- [17] Huang, G. B., Chen, L., and Siew, C. K. (2006). Universal Approximation using Incremental Constructive Feedforward Networks with Random Hidden Nodes. *IEEE Transactions on Neural Networks* 17, 879–892.
- [18] Kadali, K. S., and Rajaji, L. (2018). Evaluation of Energy in Wind Turbine System Using Probability Distribution. *Indonesian Journal of Electrical Engineering and Computer Science* 9, 294–298.
- [19] Kifune, H., Zadhe, M. K., and Sasaki, H. (2020). Efficiency estimation of synchronous generators for marine applications and verification with shop trial data and real ship operation data. *IEEE access* 8, 195541–195550.
- [20] Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering* 160, 3–24.
- [21] Libisch-Lehner, C. Machine learning and Hydro-climatic Information fostering Real-time Reservoir operation, Thesis, 2020.
- [22] Lowe, D., and Broomhead, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex systems* 2, 321–355.
- [23] McCulloch, W. S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5, 115–133.

- [24] *Modeling and Dynamic Behaviour of Hydropower Plants*; Nand Kishor, J. F.-A., Ed.; Modeling and Dynamic Behaviour of Hydropower Plants; The institution of engineering and technology: 2017.
- [25] Narayan, S. (1997). The generalized sigmoid activation function: Competitive supervised learning. *Information Sciences* 99, 69–82.
- [26] Peterson, C., and Röggnvaldsson, T. An introduction to artificial neural networks, Report, CERN, 1991.
- [27] Powell, M. J. D., Radial basis functions for multivariable interpolation: a review In *Algorithms for Approximation*; Clarendon Press: USA, 1987, pp 143–167.
- [28] Song, H., *Engineering fluid mechanics*; Springer: 2018; Vol. 2024.
- [29] Technology, E. Losses in Synchronous Motor – Power Stages & Efficiency of Synchronous Motors, Web Page, 2024.
- [30] Timilsina, A. B., Mulligan, S., and Bajracharya, T. R. (2018). Water vortex hydropower technology: a state-of-the-art review of developmental trends. *Clean Technologies and Environmental Policy* 20, 1737–1760.
- [31] Wu, Y., Wang, H., Zhang, B., and Du, K. L. (2012). Using Radial Basis Function Networks for Function Approximation and Classification. *International Scholarly Research Notices* 2012, 324194.
- [32] Zhou, Y., Tuo, Y., Cao, D. F., Li, C. L., and Wu, D. J. (2022). Study on Measurement of Penstock Head Loss for Hydropower Station with Multiple Units Per Penstock. *IOP Conference Series: Earth and Environmental Science* 1037, 012043.
- [33] Zou, J., Han, Y., and So, S.-S., Overview of Artificial Neural Networks In *Artificial neural networks: methods and applications*, Livingstone, D. J., Ed.; Springer: 2009; Chapter 2, pp 14–22.

Appendix A

Tables

A.1 Basic example

A.1.1 Input files

This table is the input files which was used for the basic example. Because the rotational speed is constant, it can be viewed as a 2 dimensional cut through an efficiency hill chart of an hydraulic machine. Each volume flow rate has an according efficiency factor.

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	0.00	0.000000	1	2.1
300	0.05	0.000000	1	2.1
300	0.10	0.000000	1	2.1
300	0.15	0.000000	1	2.1
300	0.20	0.000000	1	2.1
300	0.25	0.000000	1	2.1
300	0.30	0.000000	1	2.1
300	0.35	0.000000	1	2.1
300	0.40	0.000000	1	2.1
300	0.45	0.000000	1	2.1
300	0.50	0.000000	1	2.1
300	0.55	0.000000	1	2.1
300	0.60	0.000000	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	0.65	0.000000	1	2.1
300	0.70	0.000000	1	2.1
300	0.75	0.000000	1	2.1
300	0.80	0.000000	1	2.1
300	0.85	0.000000	1	2.1
300	0.90	0.000000	1	2.1
300	0.95	0.000000	1	2.1
300	1.00	0.000000	1	2.1
300	1.05	0.000000	1	2.1
300	1.10	0.000000	1	2.1
300	1.15	0.000000	1	2.1
300	1.20	0.000000	1	2.1
300	1.25	0.000000	1	2.1
300	1.30	0.000000	1	2.1
300	1.35	0.000000	1	2.1
300	1.40	0.000000	1	2.1
300	1.45	0.000000	1	2.1
300	1.50	0.000000	1	2.1
300	1.55	0.013450	1	2.1
300	1.60	0.024562	1	2.1
300	1.65	0.045080	1	2.1
300	1.70	0.055207	1	2.1
300	1.75	0.065333	1	2.1
300	1.80	0.075460	1	2.1
300	1.85	0.085587	1	2.1
300	1.90	0.095713	1	2.1
300	1.95	0.105840	1	2.1
300	2.00	0.115967	1	2.1
300	2.05	0.126093	1	2.1
300	2.10	0.136220	1	2.1
300	2.15	0.146347	1	2.1
300	2.20	0.156473	1	2.1
300	2.25	0.166525	1	2.1
300	2.30	0.175200	1	2.1
300	2.35	0.183800	1	2.1
300	2.40	0.192400	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	2.45	0.201000	1	2.1
300	2.50	0.209600	1	2.1
300	2.55	0.218200	1	2.1
300	2.60	0.226800	1	2.1
300	2.65	0.235400	1	2.1
300	2.70	0.244000	1	2.1
300	2.75	0.252600	1	2.1
300	2.80	0.261200	1	2.1
300	2.85	0.269800	1	2.1
300	2.90	0.278400	1	2.1
300	2.95	0.286943	1	2.1
300	3.00	0.294889	1	2.1
300	3.05	0.302767	1	2.1
300	3.10	0.309933	1	2.1
300	3.15	0.317100	1	2.1
300	3.20	0.324267	1	2.1
300	3.25	0.331433	1	2.1
300	3.30	0.338600	1	2.1
300	3.35	0.345767	1	2.1
300	3.40	0.352933	1	2.1
300	3.45	0.360100	1	2.1
300	3.50	0.367267	1	2.1
300	3.55	0.374433	1	2.1
300	3.60	0.381600	1	2.1
300	3.65	0.388767	1	2.1
300	3.70	0.395933	1	2.1
300	3.75	0.402990	1	2.1
300	3.80	0.408901	1	2.1
300	3.85	0.414807	1	2.1
300	3.90	0.420660	1	2.1
300	3.95	0.426513	1	2.1
300	4.00	0.432367	1	2.1
300	4.05	0.438220	1	2.1
300	4.10	0.444073	1	2.1
300	4.15	0.449927	1	2.1
300	4.20	0.455780	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	4.25	0.461633	1	2.1
300	4.30	0.467487	1	2.1
300	4.35	0.473340	1	2.1
300	4.40	0.479193	1	2.1
300	4.45	0.485000	1	2.1
300	4.50	0.490324	1	2.1
300	4.55	0.495593	1	2.1
300	4.60	0.500287	1	2.1
300	4.65	0.504980	1	2.1
300	4.70	0.509673	1	2.1
300	4.75	0.514367	1	2.1
300	4.80	0.519060	1	2.1
300	4.85	0.523753	1	2.1
300	4.90	0.528447	1	2.1
300	4.95	0.533140	1	2.1
300	5.00	0.537833	1	2.1
300	5.05	0.542527	1	2.1
300	5.10	0.547220	1	2.1
300	5.15	0.551913	1	2.1
300	5.20	0.556607	1	2.1
300	5.25	0.561216	1	2.1
300	5.30	0.564953	1	2.1
300	5.35	0.568687	1	2.1
300	5.40	0.572380	1	2.1
300	5.45	0.576073	1	2.1
300	5.50	0.579767	1	2.1
300	5.55	0.583460	1	2.1
300	5.60	0.587153	1	2.1
300	5.65	0.590847	1	2.1
300	5.70	0.594540	1	2.1
300	5.75	0.598233	1	2.1
300	5.80	0.601927	1	2.1
300	5.85	0.605620	1	2.1
300	5.90	0.609313	1	2.1
300	5.95	0.612973	1	2.1
300	6.00	0.616280	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	6.05	0.619547	1	2.1
300	6.10	0.622393	1	2.1
300	6.15	0.625240	1	2.1
300	6.20	0.628087	1	2.1
300	6.25	0.630933	1	2.1
300	6.30	0.633780	1	2.1
300	6.35	0.636627	1	2.1
300	6.40	0.639473	1	2.1
300	6.45	0.642320	1	2.1
300	6.50	0.645167	1	2.1
300	6.55	0.648013	1	2.1
300	6.60	0.650860	1	2.1
300	6.65	0.653707	1	2.1
300	6.70	0.656553	1	2.1
300	6.75	0.659343	1	2.1
300	6.80	0.661533	1	2.1
300	6.85	0.663720	1	2.1
300	6.90	0.665880	1	2.1
300	6.95	0.668040	1	2.1
300	7.00	0.670200	1	2.1
300	7.05	0.672360	1	2.1
300	7.10	0.674520	1	2.1
300	7.15	0.676680	1	2.1
300	7.20	0.678840	1	2.1
300	7.25	0.681000	1	2.1
300	7.30	0.683160	1	2.1
300	7.35	0.685320	1	2.1
300	7.40	0.687480	1	2.1
300	7.45	0.689618	1	2.1
300	7.50	0.691532	1	2.1
300	7.55	0.693420	1	2.1
300	7.60	0.695040	1	2.1
300	7.65	0.696660	1	2.1
300	7.70	0.698280	1	2.1
300	7.75	0.699900	1	2.1
300	7.80	0.701520	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	7.85	0.703140	1	2.1
300	7.90	0.704760	1	2.1
300	7.95	0.706380	1	2.1
300	8.00	0.708000	1	2.1
300	8.05	0.709620	1	2.1
300	8.10	0.711240	1	2.1
300	8.15	0.712860	1	2.1
300	8.20	0.714480	1	2.1
300	8.25	0.716065	1	2.1
300	8.30	0.717290	1	2.1
300	8.35	0.718513	1	2.1
300	8.40	0.719720	1	2.1
300	8.45	0.720927	1	2.1
300	8.50	0.722133	1	2.1
300	8.55	0.723340	1	2.1
300	8.60	0.724547	1	2.1
300	8.65	0.725753	1	2.1
300	8.70	0.726960	1	2.1
300	8.75	0.728167	1	2.1
300	8.80	0.729373	1	2.1
300	8.85	0.730580	1	2.1
300	8.90	0.731787	1	2.1
300	8.95	0.732981	1	2.1
300	9.00	0.734045	1	2.1
300	9.05	0.735093	1	2.1
300	9.10	0.735987	1	2.1
300	9.15	0.736880	1	2.1
300	9.20	0.737773	1	2.1
300	9.25	0.738667	1	2.1
300	9.30	0.739560	1	2.1
300	9.35	0.740453	1	2.1
300	9.40	0.741347	1	2.1
300	9.45	0.742240	1	2.1
300	9.50	0.743133	1	2.1
300	9.55	0.744027	1	2.1
300	9.60	0.744920	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	9.65	0.745813	1	2.1
300	9.70	0.746707	1	2.1
300	9.75	0.747582	1	2.1
300	9.80	0.748271	1	2.1
300	9.85	0.748960	1	2.1
300	9.90	0.749640	1	2.1
300	9.95	0.750320	1	2.1
300	10.00	0.751000	1	2.1
300	10.05	0.751680	1	2.1
300	10.10	0.752360	1	2.1
300	10.15	0.753040	1	2.1
300	10.20	0.753720	1	2.1
300	10.25	0.754400	1	2.1
300	10.30	0.755080	1	2.1
300	10.35	0.755760	1	2.1
300	10.40	0.756440	1	2.1
300	10.45	0.757114	1	2.1
300	10.50	0.757727	1	2.1
300	10.55	0.758333	1	2.1
300	10.60	0.758867	1	2.1
300	10.65	0.759400	1	2.1
300	10.70	0.759933	1	2.1
300	10.75	0.760467	1	2.1
300	10.80	0.761000	1	2.1
300	10.85	0.761533	1	2.1
300	10.90	0.762067	1	2.1
300	10.95	0.762600	1	2.1
300	11.00	0.763133	1	2.1
300	11.05	0.763667	1	2.1
300	11.10	0.764200	1	2.1
300	11.15	0.764733	1	2.1
300	11.20	0.765267	1	2.1
300	11.25	0.765792	1	2.1
300	11.30	0.766229	1	2.1
300	11.35	0.766667	1	2.1
300	11.40	0.767100	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	11.45	0.767533	1	2.1
300	11.50	0.767967	1	2.1
300	11.55	0.768400	1	2.1
300	11.60	0.768833	1	2.1
300	11.65	0.769267	1	2.1
300	11.70	0.769700	1	2.1
300	11.75	0.770133	1	2.1
300	11.80	0.770567	1	2.1
300	11.85	0.771000	1	2.1
300	11.90	0.771433	1	2.1
300	11.95	0.771864	1	2.1
300	12.00	0.772264	1	2.1
300	12.05	0.772660	1	2.1
300	12.10	0.773020	1	2.1
300	12.15	0.773380	1	2.1
300	12.20	0.773740	1	2.1
300	12.25	0.774100	1	2.1
300	12.30	0.774460	1	2.1
300	12.35	0.774820	1	2.1
300	12.40	0.775180	1	2.1
300	12.45	0.775540	1	2.1
300	12.50	0.775900	1	2.1
300	12.55	0.776260	1	2.1
300	12.60	0.776620	1	2.1
300	12.65	0.776980	1	2.1
300	12.70	0.777340	1	2.1
300	12.75	0.777694	1	2.1
300	12.80	0.777991	1	2.1
300	12.85	0.778287	1	2.1
300	12.90	0.778580	1	2.1
300	12.95	0.778873	1	2.1
300	13.00	0.779167	1	2.1
300	13.05	0.779460	1	2.1
300	13.10	0.779753	1	2.1
300	13.15	0.780047	1	2.1
300	13.20	0.780340	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (–)	a_0 (–)	H (m)
300	13.25	0.780633	1	2.1
300	13.30	0.780927	1	2.1
300	13.35	0.781220	1	2.1
300	13.40	0.781513	1	2.1
300	13.45	0.781974	1	2.1
300	13.50	0.782085	1	2.1
300	13.55	0.781513	1	2.1
300	13.60	0.781455	1	2.1
300	13.65	0.781396	1	2.1
300	13.70	0.781337	1	2.1
300	13.75	0.781279	1	2.1
300	13.80	0.781220	1	2.1
300	13.85	0.781161	1	2.1
300	13.90	0.781103	1	2.1
300	13.95	0.781044	1	2.1
300	14.00	0.780985	1	2.1
300	14.05	0.780927	1	2.1
300	14.10	0.780868	1	2.1
300	14.15	0.780809	1	2.1
300	14.20	0.780751	1	2.1
300	14.25	0.780692	1	2.1
300	14.30	0.780633	1	2.1
300	14.35	0.780575	1	2.1
300	14.40	0.780516	1	2.1
300	14.45	0.780457	1	2.1
300	14.50	0.780399	1	2.1
300	14.55	0.780340	1	2.1
300	14.60	0.780281	1	2.1
300	14.65	0.780223	1	2.1
300	14.70	0.780164	1	2.1
300	14.75	0.780105	1	2.1
300	14.80	0.780047	1	2.1
300	14.85	0.779988	1	2.1
300	14.90	0.779929	1	2.1
300	14.95	0.779871	1	2.1
300	15.00	0.779812	1	2.1

n (rpm)	Q_{11} ($\frac{m^3}{s}$)	Eta (—)	a_0 (—)	H (m)
300	15.05	0.779753	1	2.1

Table A.1: Efficiency table for both units

The parameter file of the program only consists out of 3 entries. These are the required physical constants. This file was also supposed to contain settings of the program for users who are not familiar with Python.

Abb	Number	Unit	Description
N	2.0	(—)	Number of Units
Roh	1000.0	$\frac{kg}{m^3}$	Density
g	9.81	$\frac{m}{s^2}$	Acceleration of gravity

Table A.2: Parameter File

A.1.2 Output files

The output file contains a table with the optimal distribution and the corresponding volume flow, efficiency factors and power output for each total volume flow rate.

Q_{tot} ($\frac{m^3}{s}$)	P_{tot} (W)	η_1 (—)	η_2 (—)	q_0 (—)	q_1 (—)	η_{tot} (—)	r_0 (—)	r_1 (—)
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9998	0.0002
0.05	0.0	0.0	0.0	0.0017	0.0	0.0	0.9998	0.0002
0.1	0.0	0.0	0.0	0.0033	0.0	0.0	0.9998	0.0002
0.15	0.0	0.0	0.0	0.005	0.0	0.0	0.9998	0.0002
0.2	0.0	0.0	0.0	0.0067	0.0	0.0	0.9998	0.0002
0.25	0.0	0.0	0.0	0.0083	0.0	0.0	0.9998	0.0002
0.3	0.0	0.0	0.0	0.01	0.0	0.0	0.9998	0.0002
0.35	0.0	0.0	0.0	0.0117	0.0	0.0	0.9998	0.0002
0.4	0.0	0.0	0.0	0.0133	0.0	0.0	0.9998	0.0002
0.45	0.0	0.0	0.0	0.015	0.0	0.0	0.9998	0.0002
0.5	0.0	0.0	0.0	0.0167	0.0	0.0	0.9998	0.0002
0.55	0.0	0.0	0.0	0.0183	0.0	0.0	0.9998	0.0001
0.6	0.0	0.0	0.0	0.02	0.0	0.0	0.9999	0.0001
0.65	0.0	0.0	0.0	0.0217	0.0	0.0	0.9999	0.0001
0.7	0.0	0.0	0.0	0.0233	0.0	0.0	0.9999	0.0001

$Q_{tot} \left(\frac{m^3}{s}\right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
0.75	0.0	0.0	0.0	0.025	0.0	0.0	0.9999	0.0001
0.8	0.0	0.0	0.0	0.0267	0.0	0.0	0.9999	0.0001
0.85	0.0	0.0	0.0	0.0283	0.0	0.0	0.9999	0.0001
0.9	0.0	0.0	0.0	0.03	0.0	0.0	0.9999	0.0001
0.95	0.0	0.0	0.0	0.0317	0.0	0.0	0.9999	0.0001
1.0	0.0	0.0	0.0	0.0333	0.0	0.0	0.9999	0.0001
1.05	0.0	0.0	0.0	0.035	0.0	0.0	0.9999	0.0001
1.1	0.0	0.0	0.0	0.0367	0.0	0.0	0.9999	0.0001
1.15	0.0	0.0	0.0	0.0383	0.0	0.0	0.9999	0.0001
1.2	0.0	0.0	0.0	0.04	0.0	0.0	0.9999	0.0001
1.25	0.0	0.0	0.0	0.0417	0.0	0.0	0.9999	0.0001
1.3	0.0	0.0	0.0	0.0433	0.0	0.0	0.9999	0.0001
1.35	0.0	0.0	0.0	0.045	0.0	0.0	0.9999	0.0001
1.4	0.0	0.0	0.0	0.0467	0.0	0.0	0.9999	0.0001
1.45	0.0	0.0	0.0	0.0483	0.0	0.0	0.9999	0.0001
1.5	0.0	0.0	0.0	0.05	0.0	0.0	0.9999	0.0001
1.55	436.8575	0.0137	0.0	0.0517	0.0	0.0137	0.9999	0.0001
1.6	809.5245	0.0246	0.0	0.0533	0.0	0.0246	0.9999	0.0001
1.65	1532.3436	0.0451	0.0	0.055	0.0	0.0451	0.9999	0.0001
1.7	1941.6659	0.0554	0.0	0.0567	0.0	0.0554	0.9999	0.0001
1.75	2355.2945	0.0653	0.0	0.0583	0.0	0.0653	0.9999	0.0001
1.8	2798.1926	0.0755	0.0	0.06	0.0	0.0755	0.9999	0.0001
1.85	3271.3525	0.0858	0.0	0.0617	0.0	0.0858	0.9999	0.0001
1.9	3746.3027	0.0957	0.0	0.0633	0.0	0.0957	0.9999	0.0001
1.95	4251.7992	0.1058	0.0	0.065	0.0	0.1058	0.9999	0.0001
2.0	4788.7966	0.1162	0.0	0.0667	0.0	0.1162	0.9999	0.0001
2.05	5325.0684	0.1261	0.0	0.0683	0.0	0.1261	0.9999	0.0001
2.1	5893.1633	0.1362	0.0	0.07	0.0	0.1362	0.9999	0.0001
2.15	6493.9982	0.1465	0.0	0.0717	0.0	0.1465	0.9999	0.0001
2.2	7091.5917	0.1565	0.0	0.0733	0.0	0.1565	0.9999	0.0001
2.25	7718.8026	0.1665	0.0	0.075	0.0	0.1665	1.0	0.0
2.3	8313.1416	0.1754	0.0	0.0767	0.0	0.1754	1.0	0.0
2.35	8898.0688	0.1838	0.0	0.0783	0.0	0.1838	1.0	0.0
2.4	9512.7178	0.1924	0.0	0.08	0.0	0.1924	1.0	0.0
2.45	10157.7881	0.2012	0.0	0.0817	0.0	0.2012	1.0	0.0
2.5	10794.7922	0.2096	0.0	0.0833	0.0	0.2096	1.0	0.0

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
2.55	11462.6024	0.2182	0.0	0.085	0.0	0.2182	1.0	0.0
2.6	12161.8863	0.227	0.0	0.0867	0.0	0.227	1.0	0.0
2.65	12850.9674	0.2354	0.0	0.0883	0.0	0.2354	1.0	0.0
2.7	13571.9388	0.244	0.0	0.09	0.0	0.244	1.0	0.0
2.75	14325.4363	0.2528	0.0	0.0917	0.0	0.2528	1.0	0.0
2.8	15066.5943	0.2612	0.0	0.0933	0.0	0.2612	1.0	0.0
2.85	15840.7269	0.2698	0.0	0.095	0.0	0.2698	1.0	0.0
2.9	16648.3697	0.2786	0.0	0.0967	0.0	0.2786	1.0	0.0
2.95	17438.1989	0.2869	0.0	0.0983	0.0	0.2869	1.0	0.0
3.0	18225.0132	0.2949	0.0	0.1	0.0	0.2949	1.0	0.0
3.05	19038.9994	0.3029	0.0	0.1017	0.0	0.3029	1.0	0.0
3.1	19793.1481	0.3099	0.0	0.1033	0.0	0.3099	1.0	0.0
3.15	20577.6179	0.3171	0.0	0.105	0.0	0.3171	1.0	0.0
3.2	21392.8285	0.3244	0.0	0.1067	0.0	0.3244	1.0	0.0
3.25	22190.3746	0.3314	0.0	0.1083	0.0	0.3314	1.0	0.0
3.3	23019.1454	0.3386	0.0	0.11	0.0	0.3386	1.0	0.0
3.35	23879.534	0.3459	0.0	0.1117	0.0	0.3459	1.0	0.0
3.4	24720.4775	0.3529	0.0	0.1133	0.0	0.3529	1.0	0.0
3.45	25593.5493	0.3601	0.0	0.115	0.0	0.3601	1.0	0.0
3.5	26499.1159	0.3674	0.0	0.1167	0.0	0.3674	1.0	0.0
3.55	27383.4569	0.3744	0.0	0.1183	0.0	0.3744	1.0	0.0
3.6	28300.8298	0.3816	0.0	0.12	0.0	0.3816	1.0	0.0
3.65	29251.5744	0.3889	0.0	0.1217	0.0	0.3889	1.0	0.0
3.7	30179.3128	0.3959	0.0	0.1233	0.0	0.3959	1.0	0.0
3.75	31132.5042	0.403	0.0	0.125	0.0	0.403	1.0	0.0
3.8	32027.9923	0.409	0.0	0.1267	0.0	0.409	1.0	0.0
3.85	32899.7346	0.4148	0.0	0.1283	0.0	0.4148	1.0	0.0
3.9	33797.465	0.4207	0.0	0.13	0.0	0.4207	1.0	0.0
3.95	34725.3899	0.4266	0.0	0.1317	0.0	0.4266	1.0	0.0
4.0	35628.5573	0.4324	0.0	0.1333	0.0	0.4324	1.0	0.0
4.05	36562.4694	0.4382	0.0	0.135	0.0	0.4382	1.0	0.0
4.1	37527.2931	0.4442	0.0	0.1367	0.0	0.4442	1.0	0.0
4.15	38465.9052	0.4499	0.0	0.1383	0.0	0.4499	1.0	0.0
4.2	39435.9999	0.4558	0.0	0.14	0.0	0.4558	1.0	0.0
4.25	40437.7225	0.4618	0.0	0.1417	0.0	0.4617	1.0	0.0
4.3	41411.7791	0.4675	0.0	0.1433	0.0	0.4675	1.0	0.0

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
4.35	42418.0564	0.4733	0.0	0.145	0.0	0.4733	1.0	0.0
4.4	43456.594	0.4793	0.0	0.1467	0.0	0.4793	1.0	0.0
4.45	44461.9379	0.485	0.0	0.1483	0.0	0.485	1.0	0.0
4.5	45455.2814	0.4903	0.0	0.15	0.0	0.4903	1.0	0.0
4.55	46473.2283	0.4957	0.0	0.1517	0.0	0.4957	1.0	0.0
4.6	47409.2738	0.5003	0.0	0.1533	0.0	0.5003	1.0	0.0
4.65	48374.3824	0.505	0.0	0.155	0.0	0.505	1.0	0.0
4.7	49368.5579	0.5098	0.0	0.1567	0.0	0.5098	1.0	0.0
4.75	50333.0238	0.5144	0.0	0.1583	0.0	0.5144	1.0	0.0
4.8	51327.1443	0.5191	0.0	0.16	0.0	0.5191	1.0	0.0
4.85	52350.9062	0.5238	0.0	0.1617	0.0	0.5238	1.0	0.0
4.9	53343.7923	0.5284	0.0	0.1633	0.0	0.5284	1.0	0.0
4.95	54366.9248	0.5331	0.0	0.165	0.0	0.5331	1.0	0.0
5.0	55420.2731	0.5379	0.0	0.1667	0.0	0.5379	1.0	0.0
5.05	56441.5795	0.5425	0.0	0.1683	0.0	0.5425	1.0	0.0
5.1	57493.724	0.5472	0.0	0.17	0.0	0.5472	1.0	0.0
5.15	58576.6586	0.552	0.0	0.1717	0.0	0.552	1.0	0.0
5.2	59626.3853	0.5566	0.0	0.1733	0.0	0.5566	1.0	0.0
5.25	60698.4997	0.5612	0.0	0.175	0.0	0.5612	1.0	0.0
5.3	61704.4035	0.565	0.0	0.1767	0.0	0.565	1.0	0.0
5.35	62677.8006	0.5687	0.0	0.1783	0.0	0.5687	1.0	0.0
5.4	63674.6421	0.5724	0.0	0.18	0.0	0.5724	1.0	0.0
5.45	64699.0554	0.5761	0.0	0.1817	0.0	0.5761	1.0	0.0
5.5	65690.5489	0.5798	0.0	0.1833	0.0	0.5798	1.0	0.0
5.55	66710.22	0.5835	0.0	0.185	0.0	0.5835	1.0	0.0
5.6	67757.9157	0.5872	0.0	0.1867	0.0	0.5872	1.0	0.0
5.65	68771.7741	0.5908	0.0	0.1883	0.0	0.5908	1.0	0.0
5.7	69814.2757	0.5945	0.0	0.19	0.0	0.5945	1.0	0.0
5.75	70885.2538	0.5983	0.0	0.1917	0.0	0.5983	1.0	0.0
5.8	71921.477	0.6019	0.0	0.1933	0.0	0.6019	1.0	0.0
5.85	72986.8091	0.6056	0.0	0.195	0.0	0.6056	1.0	0.0
5.9	74080.9875	0.6094	0.0	0.1967	0.0	0.6094	1.0	0.0
5.95	75135.5186	0.613	0.0	0.1983	0.0	0.613	1.0	0.0
6.0	76175.8936	0.6163	0.0	0.2	0.0	0.6163	1.0	0.0
6.05	77237.7097	0.6196	0.0	0.2017	0.0	0.6196	1.0	0.0
6.1	78213.5431	0.6224	0.0	0.2033	0.0	0.6224	1.0	0.0

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
6.15	79215.5008	0.6252	0.0	0.205	0.0	0.6252	1.0	0.0
6.2	80243.3355	0.6281	0.0	0.2067	0.0	0.6281	1.0	0.0
6.25	81236.4067	0.6309	0.0	0.2083	0.0	0.6309	1.0	0.0
6.3	82255.9612	0.6338	0.0	0.21	0.0	0.6338	1.0	0.0
6.35	83301.741	0.6367	0.0	0.2117	0.0	0.6367	1.0	0.0
6.4	84312.0501	0.6395	0.0	0.2133	0.0	0.6395	1.0	0.0
6.45	85349.2014	0.6423	0.0	0.215	0.0	0.6423	1.0	0.0
6.5	86412.9262	0.6452	0.0	0.2167	0.0	0.6452	1.0	0.0
6.55	87440.4732	0.648	0.0	0.2183	0.0	0.648	1.0	0.0
6.6	88495.2213	0.6509	0.0	0.22	0.0	0.6509	1.0	0.0
6.65	89576.8913	0.6538	0.0	0.2217	0.0	0.6538	1.0	0.0
6.7	90621.6761	0.6566	0.0	0.2233	0.0	0.6566	1.0	0.0
6.75	91686.038	0.6593	0.0	0.225	0.0	0.6593	1.0	0.0
6.8	92691.7438	0.6616	0.0	0.2267	0.0	0.6616	1.0	0.0
6.85	93661.8772	0.6637	0.0	0.2283	0.0	0.6637	1.0	0.0
6.9	94652.7778	0.6659	0.0	0.23	0.0	0.6659	1.0	0.0
6.95	95667.8781	0.6681	0.0	0.2317	0.0	0.6681	1.0	0.0
7.0	96647.331	0.6702	0.0	0.2333	0.0	0.6702	1.0	0.0
7.05	97651.5829	0.6724	0.0	0.235	0.0	0.6724	1.0	0.0
7.1	98680.2997	0.6746	0.0	0.2367	0.0	0.6746	1.0	0.0
7.15	99672.8324	0.6767	0.0	0.2383	0.0	0.6767	1.0	0.0
7.2	100690.4364	0.6788	0.0	0.24	0.0	0.6788	1.0	0.0
7.25	101732.7697	0.681	0.0	0.2417	0.0	0.681	1.0	0.0
7.3	102738.3822	0.6832	0.0	0.2433	0.0	0.6832	1.0	0.0
7.35	103769.3383	0.6853	0.0	0.245	0.0	0.6853	1.0	0.0
7.4	104825.2223	0.6875	0.0	0.2467	0.0	0.6875	1.0	0.0
7.45	105840.6748	0.6896	0.0	0.2483	0.0	0.6896	1.0	0.0
7.5	106846.8903	0.6915	0.0	0.25	0.0	0.6915	1.0	0.0
7.55	107872.1732	0.6935	0.0	0.2517	0.0	0.6935	1.0	0.0
7.6	108820.5508	0.695	0.0	0.2533	0.0	0.695	1.0	0.0
7.65	109791.9788	0.6967	0.0	0.255	0.0	0.6967	1.0	0.0
7.7	110786.0758	0.6983	0.0	0.2567	0.0	0.6983	1.0	0.0
7.75	111744.2633	0.6999	0.0	0.2583	0.0	0.6999	1.0	0.0
7.8	112725.7055	0.7015	0.0	0.26	0.0	0.7015	1.0	0.0
7.85	113730.0148	0.7032	0.0	0.2617	0.0	0.7032	1.0	0.0
7.9	114698.0121	0.7048	0.0	0.2633	0.0	0.7048	1.0	0.0

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
7.95	115689.4683	0.7064	0.0	0.265	0.0	0.7064	1.0	0.0
8.0	116703.99	0.708	0.0	0.2667	0.0	0.708	1.0	0.0
8.05	117681.7971	0.7096	0.0	0.2683	0.0	0.7096	1.0	0.0
8.1	118683.2674	0.7112	0.0	0.27	0.0	0.7112	1.0	0.0
8.15	119708.0014	0.7129	0.0	0.2717	0.0	0.7129	1.0	0.0
8.2	120695.6184	0.7145	0.0	0.2733	0.0	0.7145	1.0	0.0
8.25	121701.2297	0.7161	0.0	0.275	0.0	0.7161	1.0	0.0
8.3	122667.1885	0.7173	0.0	0.2767	0.0	0.7173	1.0	0.0
8.35	123597.288	0.7185	0.0	0.2783	0.0	0.7185	1.0	0.0
8.4	124546.3944	0.7197	0.0	0.28	0.0	0.7197	1.0	0.0
8.45	125516.8501	0.721	0.0	0.2817	0.0	0.721	1.0	0.0
8.5	126451.4938	0.7221	0.0	0.2833	0.0	0.7221	1.0	0.0
8.55	127408.0588	0.7233	0.0	0.285	0.0	0.7233	1.0	0.0
8.6	128386.1211	0.7246	0.0	0.2867	0.0	0.7246	1.0	0.0
8.65	129328.0717	0.7258	0.0	0.2883	0.0	0.7258	1.0	0.0
8.7	130292.0958	0.727	0.0	0.29	0.0	0.727	1.0	0.0
8.75	131277.7648	0.7282	0.0	0.2917	0.0	0.7282	1.0	0.0
8.8	132227.0223	0.7294	0.0	0.2933	0.0	0.7294	1.0	0.0
8.85	133198.5054	0.7306	0.0	0.295	0.0	0.7306	1.0	0.0
8.9	134191.7354	0.7318	0.0	0.2967	0.0	0.7318	1.0	0.0
8.95	135146.0415	0.733	0.0	0.2983	0.0	0.733	1.0	0.0
9.0	136098.4624	0.734	0.0	0.3	0.0	0.734	1.0	0.0
9.05	137068.5778	0.7351	0.0	0.3017	0.0	0.7351	1.0	0.0
9.1	137974.5729	0.736	0.0	0.3033	0.0	0.736	1.0	0.0
9.15	138901.2537	0.7369	0.0	0.305	0.0	0.7369	1.0	0.0
9.2	139848.1751	0.7378	0.0	0.3067	0.0	0.7378	1.0	0.0
9.25	140759.5798	0.7387	0.0	0.3083	0.0	0.7387	1.0	0.0
9.3	141691.7827	0.7396	0.0	0.31	0.0	0.7396	1.0	0.0
9.35	142644.3357	0.7405	0.0	0.3117	0.0	0.7405	1.0	0.0
9.4	143561.1499	0.7413	0.0	0.3133	0.0	0.7413	1.0	0.0
9.45	144498.875	0.7422	0.0	0.315	0.0	0.7422	1.0	0.0
9.5	145457.0594	0.7432	0.0	0.3167	0.0	0.7432	1.0	0.0
9.55	146379.2832	0.744	0.0	0.3183	0.0	0.744	1.0	0.0
9.6	147322.5304	0.7449	0.0	0.32	0.0	0.7449	1.0	0.0
9.65	148286.3464	0.7458	0.0	0.3217	0.0	0.7458	1.0	0.0
9.7	149213.9797	0.7467	0.0	0.3233	0.0	0.7467	1.0	0.0

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
9.75	150159.1667	0.7476	0.0	0.325	0.0	0.7476	1.0	0.0
9.8	151086.5769	0.7483	0.0	0.3267	0.0	0.7483	1.0	0.0
9.85	151978.6686	0.749	0.0	0.3283	0.0	0.749	1.0	0.0
9.9	152889.003	0.7496	0.0	0.33	0.0	0.7496	1.0	0.0
9.95	153818.8014	0.7503	0.0	0.3317	0.0	0.7503	1.0	0.0
10.0	154713.3273	0.751	0.0	0.3333	0.0	0.751	1.0	0.0
10.05	155627.8648	0.7517	0.0	0.335	0.0	0.7517	1.0	0.0
10.1	156561.9498	0.7524	0.0	0.3367	0.0	0.7524	1.0	0.0
10.15	157460.5934	0.753	0.0	0.3383	0.0	0.753	1.0	0.0
10.2	158379.3343	0.7537	0.0	0.34	0.0	0.7537	1.0	0.0
10.25	159317.7061	0.7544	0.0	0.3417	0.0	0.7544	1.0	0.0
10.3	160220.4673	0.7551	0.0	0.3433	0.0	0.7551	1.0	0.0
10.35	161143.4117	0.7558	0.0	0.345	0.0	0.7558	1.0	0.0
10.4	162086.045	0.7565	0.0	0.3467	0.0	0.7565	1.0	0.0
10.45	162991.6898	0.7571	0.0	0.3483	0.0	0.7571	1.0	0.0
10.5	163904.3554	0.7577	0.0	0.35	0.0	0.7577	1.0	0.0
10.55	164834.5247	0.7583	0.0	0.3517	0.0	0.7583	1.0	0.0
10.6	165713.9897	0.7589	0.0	0.3533	0.0	0.7589	1.0	0.0
10.65	166612.8536	0.7594	0.0	0.355	0.0	0.7594	1.0	0.0
10.7	167530.6435	0.7599	0.0	0.3567	0.0	0.7599	1.0	0.0
10.75	168413.3381	0.7605	0.0	0.3583	0.0	0.7605	1.0	0.0
10.8	169315.4988	0.761	0.0	0.36	0.0	0.761	1.0	0.0
10.85	170236.6508	0.7615	0.0	0.3617	0.0	0.7615	1.0	0.0
10.9	171122.5749	0.7621	0.0	0.3633	0.0	0.7621	1.0	0.0
10.95	172028.0325	0.7626	0.0	0.365	0.0	0.7626	1.0	0.0
11.0	172952.5465	0.7631	0.0	0.3667	0.0	0.7631	1.0	0.0
11.05	173841.7002	0.7637	0.0	0.3683	0.0	0.7637	1.0	0.0
11.1	174750.4546	0.7642	0.0	0.37	0.0	0.7642	1.0	0.0
11.15	175678.3307	0.7647	0.0	0.3717	0.0	0.7647	1.0	0.0
11.2	176570.7141	0.7653	0.0	0.3733	0.0	0.7653	1.0	0.0
11.25	177480.8276	0.7658	0.0	0.375	0.0	0.7658	1.0	0.0
11.3	178389.3465	0.7662	0.0	0.3767	0.0	0.7662	1.0	0.0
11.35	179262.8566	0.7667	0.0	0.3783	0.0	0.7667	1.0	0.0
11.4	180154.5089	0.7671	0.0	0.38	0.0	0.7671	1.0	0.0
11.45	181064.732	0.7675	0.0	0.3817	0.0	0.7675	1.0	0.0
11.5	181939.9562	0.768	0.0	0.3833	0.0	0.768	1.0	0.0

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
11.55	182834.287	0.7684	0.0	0.385	0.0	0.7684	1.0	0.0
11.6	183747.2418	0.7688	0.0	0.3867	0.0	0.7688	1.0	0.0
11.65	184625.09	0.7693	0.0	0.3883	0.0	0.7693	1.0	0.0
11.7	185522.0995	0.7697	0.0	0.39	0.0	0.7697	1.0	0.0
11.75	186437.786	0.7701	0.0	0.3917	0.0	0.7701	1.0	0.0
11.8	187318.2582	0.7706	0.0	0.3933	0.0	0.7706	1.0	0.0
11.85	188217.9464	0.771	0.0	0.395	0.0	0.771	1.0	0.0
11.9	189136.3502	0.7714	0.0	0.3967	0.0	0.7714	1.0	0.0
11.95	190018.7408	0.7719	0.0	0.3983	0.0	0.7719	1.0	0.0
12.0	190912.8324	0.7723	0.0	0.4	0.0	0.7723	1.0	0.0
12.05	191824.4074	0.7727	0.0	0.4017	0.0	0.7727	1.0	0.0
12.1	192692.1415	0.773	0.0	0.4033	0.0	0.773	1.0	0.0
12.15	193578.6768	0.7734	0.0	0.405	0.0	0.7734	1.0	0.0
12.2	194483.526	0.7737	0.0	0.4067	0.0	0.7737	1.0	0.0
12.25	195353.4401	0.7741	0.0	0.4083	0.0	0.7741	1.0	0.0
12.3	196242.2007	0.7745	0.0	0.41	0.0	0.7745	1.0	0.0
12.35	197149.3193	0.7748	0.0	0.4117	0.0	0.7748	1.0	0.0
12.4	198021.4133	0.7752	0.0	0.4133	0.0	0.7752	1.0	0.0
12.45	198912.3993	0.7755	0.0	0.415	0.0	0.7755	1.0	0.0
12.5	199821.7873	0.7759	0.0	0.4167	0.0	0.7759	1.0	0.0
12.55	200696.0613	0.7763	0.0	0.4183	0.0	0.7763	1.0	0.0
12.6	201589.2726	0.7766	0.0	0.42	0.0	0.7766	1.0	0.0
12.65	202500.93	0.777	0.0	0.4217	0.0	0.777	1.0	0.0
12.7	203377.384	0.7773	0.0	0.4233	0.0	0.7773	1.0	0.0
12.75	204271.3567	0.7777	0.0	0.425	0.0	0.7777	1.0	0.0
12.8	205168.1277	0.778	0.0	0.4267	0.0	0.778	1.0	0.0
12.85	206030.0885	0.7783	0.0	0.4283	0.0	0.7783	1.0	0.0
12.9	206909.8929	0.7786	0.0	0.43	0.0	0.7786	1.0	0.0
12.95	207807.7364	0.7789	0.0	0.4317	0.0	0.7789	1.0	0.0
13.0	208670.7863	0.7792	0.0	0.4333	0.0	0.7792	1.0	0.0
13.05	209552.4038	0.7795	0.0	0.435	0.0	0.7795	1.0	0.0
13.1	210452.0964	0.7798	0.0	0.4367	0.0	0.7798	1.0	0.0
13.15	211316.9226	0.78	0.0	0.4383	0.0	0.78	1.0	0.0
13.2	212200.3533	0.7803	0.0	0.44	0.0	0.7803	1.0	0.0
13.25	213101.8951	0.7806	0.0	0.4417	0.0	0.7806	1.0	0.0
13.3	213968.4975	0.7809	0.0	0.4433	0.0	0.7809	1.0	0.0

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
13.35	214853.7415	0.7812	0.0	0.445	0.0	0.7812	1.0	0.0
13.4	215758.0563	0.7815	0.0	0.4467	0.0	0.7815	1.0	0.0
13.45	216671.8671	0.782	0.0	0.4483	0.0	0.782	1.0	0.0
13.5	217508.4523	0.7821	0.0	0.45	0.0	0.7821	1.0	0.0
13.55	218170.1786	0.7815	0.0	0.4517	0.0	0.7815	1.0	0.0
13.6	218942.8095	0.7815	0.0	0.4533	0.0	0.7815	1.0	0.0
13.65	219731.4073	0.7814	0.0	0.455	0.0	0.7814	1.0	0.0
13.7	220535.4917	0.7813	0.0	0.4567	0.0	0.7813	1.0	0.0
13.75	221307.7673	0.7813	0.0	0.4583	0.0	0.7813	1.0	0.0
13.8	222096.0024	0.7812	0.0	0.46	0.0	0.7812	1.0	0.0
13.85	222899.717	0.7812	0.0	0.4617	0.0	0.7812	1.0	0.0
13.9	223671.6373	0.7811	0.0	0.4633	0.0	0.7811	1.0	0.0
13.95	224459.5098	0.781	0.0	0.465	0.0	0.781	1.0	0.0
14.0	225262.8546	0.781	0.0	0.4667	0.0	0.781	1.0	0.0
14.05	226034.4197	0.7809	0.0	0.4683	0.0	0.7809	1.0	0.0
14.1	226821.9295	0.7809	0.0	0.47	0.0	0.7809	1.0	0.0
14.15	227624.9044	0.7808	0.0	0.4717	0.0	0.7808	1.0	0.0
14.2	228396.1143	0.7808	0.0	0.4733	0.0	0.7808	1.0	0.0
14.25	229183.2615	0.7807	0.0	0.475	0.0	0.7807	1.0	0.0
14.3	229985.8665	0.7806	0.0	0.4767	0.0	0.7806	1.0	0.0
14.35	230756.7211	0.7806	0.0	0.4783	0.0	0.7806	1.0	0.0
14.4	231543.5057	0.7805	0.0	0.48	0.0	0.7805	1.0	0.0
14.45	232345.7409	0.7805	0.0	0.4817	0.0	0.7805	1.0	0.0
14.5	233116.2402	0.7804	0.0	0.4833	0.0	0.7804	1.0	0.0
14.55	233902.6621	0.7803	0.0	0.485	0.0	0.7803	1.0	0.0
14.6	234704.5276	0.7803	0.0	0.4867	0.0	0.7803	1.0	0.0
14.65	235474.6716	0.7802	0.0	0.4883	0.0	0.7802	1.0	0.0
14.7	236260.7309	0.7802	0.0	0.49	0.0	0.7802	1.0	0.0
14.75	237062.2265	0.7801	0.0	0.4917	0.0	0.7801	1.0	0.0
14.8	237832.0153	0.78	0.0	0.4933	0.0	0.78	1.0	0.0
14.85	238617.7119	0.78	0.0	0.495	0.0	0.78	1.0	0.0
14.9	239418.8377	0.7799	0.0	0.4967	0.0	0.7799	1.0	0.0
14.95	240188.2712	0.7799	0.0	0.4983	0.0	0.7799	1.0	0.0
15.0	240973.6052	0.7798	0.0	0.5	0.0	0.7798	1.0	0.0
15.05	240973.6052	0.7798	0.0	0.5	0.0017	0.7798	1.0	0.0
15.1	240973.6052	0.7798	0.0	0.5	0.0033	0.7798	1.0	0.0

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
15.15	240973.6052	0.7798	0.0	0.5	0.005	0.7798	1.0	0.0
15.2	240973.6052	0.7798	0.0	0.5	0.0067	0.7798	1.0	0.0
15.25	240973.6052	0.7798	0.0	0.5	0.0083	0.7798	1.0	0.0
15.3	240973.6052	0.7798	0.0	0.5	0.01	0.7798	1.0	0.0
15.35	240973.6052	0.7798	0.0	0.5	0.0117	0.7798	1.0	0.0
15.4	240973.6052	0.7798	0.0	0.5	0.0133	0.7798	1.0	0.0
15.45	240973.6052	0.7798	0.0	0.5	0.015	0.7798	1.0	0.0
15.5	240973.6052	0.7798	0.0	0.5	0.0167	0.7798	1.0	0.0
15.55	240973.6052	0.7798	0.0	0.5	0.0183	0.7798	1.0	0.0
15.6	240973.6052	0.7798	0.0	0.5	0.02	0.7798	1.0	0.0
15.65	240973.6052	0.7798	0.0	0.5	0.0217	0.7798	1.0	0.0
15.7	240973.6052	0.7798	0.0	0.5	0.0233	0.7798	1.0	0.0
15.75	240973.6052	0.7798	0.0	0.5	0.025	0.7798	1.0	0.0
15.8	240973.6052	0.7798	0.0	0.5	0.0267	0.7798	0.9999	0.0001
15.85	240973.6052	0.7798	0.0	0.5	0.0283	0.7797	0.9999	0.0001
15.9	240973.6052	0.7798	0.0	0.5	0.03	0.7797	0.9999	0.0001
15.95	240973.6052	0.7798	0.0	0.5	0.0317	0.7797	0.9998	0.0002
16.0	240973.6052	0.7798	0.0	0.5	0.0333	0.7796	0.9997	0.0003
16.05	240973.6052	0.7798	0.0	0.5	0.035	0.7795	0.9996	0.0004
16.1	240973.6052	0.7798	0.0	0.5	0.0367	0.7793	0.9994	0.0006
16.15	240973.6052	0.7798	0.0	0.5	0.0383	0.779	0.999	0.001
16.2	240973.6052	0.7798	0.0	0.5	0.04	0.7783	0.998	0.002
16.25	240973.6052	0.7798	0.0	0.5	0.0417	0.7761	0.9952	0.0048
16.3	240973.6052	0.7798	0.0	0.5	0.0433	0.7669	0.9835	0.0165
16.35	240973.6052	0.7798	0.0	0.5	0.045	0.7154	0.9174	0.0826
16.4	240070.729	0.7438	0.6642	0.318	0.2287	0.7105	0.5817	0.4183
16.45	242436.4512	0.7153	0.7153	0.2742	0.2742	0.7153	0.5	0.5
16.5	243402.4594	0.7161	0.7161	0.275	0.275	0.7161	0.5	0.5
16.55	244349.0176	0.7167	0.7167	0.2759	0.2758	0.7167	0.5	0.5
16.6	245334.377	0.7173	0.7173	0.2767	0.2767	0.7173	0.5	0.5
16.65	246245.0763	0.7179	0.7179	0.2775	0.2775	0.7179	0.5	0.5
16.7	247195.0899	0.7185	0.7185	0.2784	0.2783	0.7185	0.5	0.5
16.75	248181.2087	0.7191	0.7191	0.2792	0.2792	0.7191	0.5	0.5
16.8	249092.7889	0.7197	0.7197	0.28	0.28	0.7197	0.5	0.5
16.85	250043.761	0.7203	0.7203	0.2809	0.2808	0.7203	0.5	0.5
16.9	251033.7002	0.721	0.721	0.2817	0.2817	0.721	0.5	0.5

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
16.95	251976.2062	0.7275	0.7154	0.2907	0.2743	0.7216	0.5145	0.4855
17.0	252964.5443	0.7273	0.7169	0.2905	0.2762	0.7223	0.5126	0.4874
17.05	253897.378	0.7228	0.7228	0.2842	0.2842	0.7228	0.5	0.5
17.1	254816.1175	0.7233	0.7233	0.285	0.285	0.7233	0.5	0.5
17.15	255774.5486	0.724	0.7239	0.2859	0.2858	0.7239	0.5	0.5
17.2	256772.2422	0.7246	0.7246	0.2867	0.2867	0.7246	0.5	0.5
17.25	257694.5613	0.7251	0.7251	0.2875	0.2875	0.7251	0.5	0.5
17.3	258656.722	0.7258	0.7257	0.2884	0.2883	0.7258	0.5	0.5
17.35	259658.2928	0.7264	0.7264	0.2892	0.2892	0.7264	0.5	0.5
17.4	260584.1915	0.727	0.727	0.29	0.29	0.727	0.5	0.5
17.45	261550.0817	0.7276	0.7275	0.2909	0.2908	0.7276	0.5	0.5
17.5	262555.5297	0.7282	0.7282	0.2917	0.2917	0.7282	0.5	0.5
17.55	263485.008	0.7288	0.7288	0.2925	0.2925	0.7288	0.5	0.5
17.6	264454.6277	0.7294	0.7293	0.2934	0.2933	0.7294	0.5	0.5
17.65	265463.9529	0.73	0.73	0.2942	0.2942	0.73	0.5	0.5
17.7	266397.0109	0.7306	0.7306	0.295	0.295	0.7306	0.5	0.5
17.75	267370.3601	0.7312	0.7312	0.2959	0.2958	0.7312	0.5	0.5
17.8	268383.4708	0.7318	0.7318	0.2967	0.2967	0.7318	0.5	0.5
17.85	269317.9019	0.7324	0.7324	0.2975	0.2975	0.7324	0.5	0.5
17.9	270292.1832	0.733	0.733	0.2984	0.2983	0.733	0.5	0.5
17.95	271282.2411	0.7335	0.7335	0.2992	0.2992	0.7335	0.5	0.5
18.0	272196.9248	0.734	0.734	0.3	0.3	0.734	0.5	0.5
18.05	273148.2162	0.7346	0.7345	0.3009	0.3008	0.7346	0.5	0.5
18.1	274137.1556	0.7351	0.7351	0.3017	0.3017	0.7351	0.5	0.5
18.15	275024.4007	0.7355	0.7355	0.3025	0.3025	0.7355	0.5	0.5
18.2	275949.7019	0.736	0.736	0.3034	0.3033	0.736	0.5	0.5
18.25	276912.6121	0.7365	0.7365	0.3042	0.3042	0.7365	0.5	0.5
18.3	277802.5073	0.7369	0.7369	0.305	0.305	0.7369	0.5	0.5
18.35	278730.5696	0.7373	0.7373	0.3059	0.3058	0.7373	0.5	0.5
18.4	279696.3503	0.7378	0.7378	0.3067	0.3067	0.7378	0.5	0.5
18.45	280588.8956	0.7382	0.7382	0.3075	0.3075	0.7382	0.5	0.5
18.5	281519.719	0.7387	0.7386	0.3084	0.3083	0.7387	0.5	0.5
18.55	282488.37	0.7391	0.7391	0.3092	0.3092	0.7391	0.5	0.5
18.6	283383.5654	0.7396	0.7396	0.31	0.31	0.7396	0.5	0.5
18.65	284317.1499	0.74	0.74	0.3109	0.3108	0.74	0.5	0.5
18.7	285288.6714	0.7405	0.7405	0.3117	0.3117	0.7405	0.5	0.5

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
18.75	286186.5169	0.7409	0.7409	0.3125	0.3125	0.7409	0.5	0.5
18.8	287122.8625	0.7414	0.7413	0.3134	0.3133	0.7413	0.5	0.5
18.85	288097.2543	0.7418	0.7418	0.3142	0.3142	0.7418	0.5	0.5
18.9	288997.7499	0.7422	0.7422	0.315	0.315	0.7422	0.5	0.5
18.95	289936.8566	0.7427	0.7427	0.3159	0.3158	0.7427	0.5	0.5
19.0	290914.1189	0.7432	0.7432	0.3167	0.3167	0.7432	0.5	0.5
19.05	291817.2646	0.7436	0.7436	0.3175	0.3175	0.7436	0.5	0.5
19.1	292759.1323	0.744	0.744	0.3184	0.3183	0.744	0.5	0.5
19.15	293739.265	0.7445	0.7445	0.3192	0.3192	0.7445	0.5	0.5
19.2	294645.0609	0.7449	0.7449	0.32	0.32	0.7449	0.5	0.5
19.25	295589.6897	0.7454	0.7453	0.3209	0.3208	0.7454	0.5	0.5
19.3	296572.6928	0.7458	0.7458	0.3217	0.3217	0.7458	0.5	0.5
19.35	297481.1387	0.7463	0.7463	0.3225	0.3225	0.7463	0.5	0.5
19.4	298428.4567	0.7467	0.7467	0.3234	0.3233	0.7467	0.5	0.5
19.45	299410.6856	0.7472	0.7472	0.3242	0.3242	0.7472	0.5	0.5
19.5	300318.3334	0.7476	0.7476	0.325	0.325	0.7476	0.5	0.5
19.55	301227.3755	0.7479	0.7479	0.3259	0.3258	0.7479	0.5	0.5
19.6	302173.1539	0.7483	0.7483	0.3267	0.3267	0.7483	0.5	0.5
19.65	303046.8777	0.7486	0.7486	0.3275	0.3275	0.7486	0.5	0.5
19.7	303957.8496	0.749	0.7489	0.3284	0.3283	0.749	0.5	0.5
19.75	304903.9103	0.7493	0.7493	0.3292	0.3292	0.7493	0.5	0.5
19.8	305778.0061	0.7496	0.7496	0.33	0.33	0.7496	0.5	0.5
19.85	306689.3918	0.75	0.75	0.3309	0.3308	0.75	0.5	0.5
19.9	307637.6029	0.7503	0.7503	0.3317	0.3317	0.7503	0.5	0.5
19.95	308513.7159	0.7507	0.7507	0.3325	0.3325	0.7507	0.5	0.5
20.0	309427.2033	0.751	0.751	0.3334	0.3333	0.751	0.5	0.5
20.05	310377.5993	0.7514	0.7514	0.3342	0.3342	0.7514	0.5	0.5
20.1	311255.7296	0.7517	0.7517	0.335	0.335	0.7517	0.5	0.5
20.15	312171.3187	0.752	0.752	0.3359	0.3358	0.752	0.5	0.5
20.2	313123.8997	0.7524	0.7524	0.3367	0.3367	0.7524	0.5	0.5
20.25	314004.0472	0.7527	0.7527	0.3375	0.3375	0.7527	0.5	0.5
20.3	314921.738	0.7531	0.753	0.3384	0.3383	0.753	0.5	0.5
20.35	315876.5039	0.7534	0.7534	0.3392	0.3392	0.7534	0.5	0.5
20.4	316758.6687	0.7537	0.7537	0.34	0.34	0.7537	0.5	0.5
20.45	317678.4613	0.7541	0.754	0.3409	0.3408	0.7541	0.5	0.5
20.5	318635.4121	0.7544	0.7544	0.3417	0.3417	0.7544	0.5	0.5

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
20.55	319519.5941	0.7547	0.7547	0.3425	0.3425	0.7547	0.5	0.5
20.6	320441.4884	0.7551	0.7551	0.3434	0.3433	0.7551	0.5	0.5
20.65	321400.6242	0.7554	0.7554	0.3442	0.3442	0.7554	0.5	0.5
20.7	322286.8234	0.7558	0.7558	0.345	0.345	0.7558	0.5	0.5
20.75	323210.8195	0.7561	0.7561	0.3459	0.3458	0.7561	0.5	0.5
20.8	324172.09	0.7565	0.7565	0.3467	0.3467	0.7565	0.5	0.5
20.85	325059.1001	0.7568	0.7568	0.3475	0.3475	0.7568	0.5	0.5
20.9	325983.6695	0.7571	0.7571	0.3484	0.3483	0.7571	0.5	0.5
20.95	326932.4141	0.7574	0.7574	0.3492	0.3492	0.7574	0.5	0.5
21.0	327808.7107	0.7577	0.7577	0.35	0.35	0.7577	0.5	0.5
21.05	328720.8099	0.758	0.758	0.3509	0.3508	0.758	0.5	0.5
21.1	329669.0494	0.7583	0.7583	0.3517	0.3517	0.7583	0.5	0.5
21.15	330530.4784	0.7586	0.7586	0.3525	0.3525	0.7586	0.5	0.5
21.2	331428.5187	0.7589	0.7589	0.3534	0.3533	0.7589	0.5	0.5
21.25	332362.6961	0.7591	0.7591	0.3542	0.3542	0.7591	0.5	0.5
21.3	333225.7072	0.7594	0.7594	0.355	0.355	0.7594	0.5	0.5
21.35	334125.396	0.7597	0.7597	0.3559	0.3558	0.7597	0.5	0.5
21.4	335061.287	0.7599	0.7599	0.3567	0.3567	0.7599	0.5	0.5
21.45	335925.8803	0.7602	0.7602	0.3575	0.3575	0.7602	0.5	0.5
21.5	336827.2174	0.7605	0.7605	0.3584	0.3583	0.7605	0.5	0.5
21.55	337764.8221	0.7607	0.7607	0.3592	0.3592	0.7607	0.5	0.5
21.6	338630.9976	0.761	0.761	0.36	0.36	0.761	0.5	0.5
21.65	339533.9831	0.7613	0.7613	0.3609	0.3608	0.7613	0.5	0.5
21.7	340473.3015	0.7615	0.7615	0.3617	0.3617	0.7615	0.5	0.5
21.75	341341.0592	0.7618	0.7618	0.3625	0.3625	0.7618	0.5	0.5
21.8	342245.6931	0.7621	0.7621	0.3634	0.3633	0.7621	0.5	0.5
21.85	343186.7252	0.7623	0.7623	0.3642	0.3642	0.7623	0.5	0.5
21.9	344056.0649	0.7626	0.7626	0.365	0.365	0.7626	0.5	0.5
21.95	344962.3473	0.7629	0.7629	0.3659	0.3658	0.7629	0.5	0.5
22.0	345905.093	0.7631	0.7631	0.3667	0.3667	0.7631	0.5	0.5
22.05	346776.015	0.7634	0.7634	0.3675	0.3675	0.7634	0.5	0.5
22.1	347683.9457	0.7637	0.7637	0.3684	0.3683	0.7637	0.5	0.5
22.15	348628.4051	0.7639	0.7639	0.3692	0.3692	0.7639	0.5	0.5
22.2	349500.9092	0.7642	0.7642	0.37	0.37	0.7642	0.5	0.5
22.25	350410.4884	0.7645	0.7645	0.3709	0.3708	0.7645	0.5	0.5
22.3	351356.6615	0.7647	0.7647	0.3717	0.3717	0.7647	0.5	0.5

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
22.35	352230.7478	0.765	0.765	0.3725	0.3725	0.765	0.5	0.5
22.4	353141.9364	0.7653	0.7653	0.3734	0.3733	0.7653	0.5	0.5
22.45	354087.8513	0.7655	0.7655	0.3742	0.3742	0.7655	0.5	0.5
22.5	354961.6553	0.7658	0.7658	0.375	0.375	0.7658	0.5	0.5
22.55	355852.3059	0.766	0.766	0.3759	0.3758	0.766	0.5	0.5
22.6	356778.693	0.7662	0.7662	0.3767	0.3767	0.7662	0.5	0.5
22.65	357634.3349	0.7664	0.7664	0.3775	0.3775	0.7664	0.5	0.5
22.7	358526.2299	0.7667	0.7667	0.3784	0.3783	0.7667	0.5	0.5
22.75	359452.9801	0.7669	0.7669	0.3792	0.3792	0.7669	0.5	0.5
22.8	360309.0179	0.7671	0.7671	0.38	0.38	0.7671	0.5	0.5
22.85	361201.3402	0.7673	0.7673	0.3809	0.3808	0.7673	0.5	0.5
22.9	362129.4641	0.7675	0.7675	0.3817	0.3817	0.7675	0.5	0.5
22.95	362986.7874	0.7678	0.7678	0.3825	0.3825	0.7678	0.5	0.5
23.0	363880.449	0.768	0.768	0.3834	0.3833	0.768	0.5	0.5
23.05	364809.9652	0.7682	0.7682	0.3842	0.3842	0.7682	0.5	0.5
23.1	365668.574	0.7684	0.7684	0.385	0.385	0.7684	0.5	0.5
23.15	366563.575	0.7686	0.7686	0.3859	0.3858	0.7686	0.5	0.5
23.2	367494.4836	0.7688	0.7688	0.3867	0.3867	0.7688	0.5	0.5
23.25	368354.3779	0.769	0.769	0.3875	0.3875	0.769	0.5	0.5
23.3	369250.7182	0.7693	0.7693	0.3884	0.3883	0.7693	0.5	0.5
23.35	370183.0192	0.7695	0.7695	0.3892	0.3892	0.7695	0.5	0.5
23.4	371044.199	0.7697	0.7697	0.39	0.39	0.7697	0.5	0.5
23.45	371941.8786	0.7699	0.7699	0.3909	0.3908	0.7699	0.5	0.5
23.5	372875.5719	0.7701	0.7701	0.3917	0.3917	0.7701	0.5	0.5
23.55	373738.0372	0.7704	0.7704	0.3925	0.3925	0.7704	0.5	0.5
23.6	374637.0562	0.7706	0.7706	0.3934	0.3933	0.7706	0.5	0.5
23.65	375572.1419	0.7708	0.7708	0.3942	0.3942	0.7708	0.5	0.5
23.7	376435.8927	0.771	0.771	0.395	0.395	0.771	0.5	0.5
23.75	377336.251	0.7712	0.7712	0.3959	0.3958	0.7712	0.5	0.5
23.8	378272.7003	0.7714	0.7714	0.3967	0.3967	0.7714	0.5	0.5
23.85	379137.0467	0.7716	0.7716	0.3975	0.3975	0.7716	0.5	0.5
23.9	380037.8706	0.7719	0.7719	0.3984	0.3983	0.7719	0.5	0.5
23.95	380967.3043	0.7721	0.7721	0.3992	0.3992	0.7721	0.5	0.5
24.0	381825.6649	0.7723	0.7723	0.4	0.4	0.7723	0.5	0.5
24.05	382719.5097	0.7725	0.7725	0.4009	0.4008	0.7725	0.5	0.5
24.1	383648.8148	0.7727	0.7727	0.4017	0.4017	0.7727	0.5	0.5

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
24.15	384498.8357	0.7728	0.7728	0.4025	0.4025	0.7728	0.5	0.5
24.2	385384.815	0.773	0.773	0.4034	0.4033	0.773	0.5	0.5
24.25	386306.2647	0.7732	0.7732	0.4042	0.4042	0.7732	0.5	0.5
24.3	387157.3535	0.7734	0.7734	0.405	0.405	0.7734	0.5	0.5
24.35	388044.4455	0.7736	0.7736	0.4059	0.4058	0.7736	0.5	0.5
24.4	388967.052	0.7737	0.7737	0.4067	0.4067	0.7737	0.5	0.5
24.45	389819.2087	0.7739	0.7739	0.4075	0.4075	0.7739	0.5	0.5
24.5	390707.4134	0.7741	0.7741	0.4084	0.4083	0.7741	0.5	0.5
24.55	391631.1766	0.7743	0.7743	0.4092	0.4092	0.7743	0.5	0.5
24.6	392484.4013	0.7745	0.7745	0.41	0.41	0.7745	0.5	0.5
24.65	393373.7186	0.7746	0.7746	0.4109	0.4108	0.7746	0.5	0.5
24.7	394298.6386	0.7748	0.7748	0.4117	0.4117	0.7748	0.5	0.5
24.75	395152.9312	0.775	0.775	0.4125	0.4125	0.775	0.5	0.5
24.8	396043.3613	0.7752	0.7752	0.4134	0.4133	0.7752	0.5	0.5
24.85	396969.4379	0.7754	0.7754	0.4142	0.4142	0.7754	0.5	0.5
24.9	397824.7985	0.7755	0.7755	0.415	0.415	0.7755	0.5	0.5
24.95	398716.3412	0.7757	0.7757	0.4159	0.4158	0.7757	0.5	0.5
25.0	399643.5746	0.7759	0.7759	0.4167	0.4167	0.7759	0.5	0.5
25.05	400500.0032	0.7761	0.7761	0.4175	0.4175	0.7761	0.5	0.5
25.1	401392.6586	0.7763	0.7763	0.4184	0.4183	0.7763	0.5	0.5
25.15	402321.0487	0.7764	0.7764	0.4192	0.4192	0.7764	0.5	0.5
25.2	403178.5452	0.7766	0.7766	0.42	0.42	0.7766	0.5	0.5
25.25	404072.3133	0.7768	0.7768	0.4209	0.4208	0.7768	0.5	0.5
25.3	405001.8601	0.777	0.777	0.4217	0.4217	0.777	0.5	0.5
25.35	405860.4246	0.7772	0.7772	0.4225	0.4225	0.7772	0.5	0.5
25.4	406755.2759	0.7773	0.7773	0.4234	0.4233	0.7773	0.5	0.5
25.45	407684.4892	0.7775	0.7775	0.4242	0.4242	0.7775	0.5	0.5
25.5	408542.7134	0.7777	0.7777	0.425	0.425	0.7777	0.5	0.5
25.55	409421.9194	0.7778	0.7778	0.4259	0.4258	0.7778	0.5	0.5
25.6	410336.2554	0.778	0.778	0.4267	0.4267	0.778	0.5	0.5
25.65	411180.6513	0.7781	0.7781	0.4275	0.4275	0.7781	0.5	0.5
25.7	412060.6911	0.7783	0.7783	0.4284	0.4283	0.7783	0.5	0.5
25.75	412975.1917	0.7784	0.7784	0.4292	0.4292	0.7784	0.5	0.5
25.8	413819.7858	0.7786	0.7786	0.43	0.43	0.7786	0.5	0.5
25.85	414700.0436	0.7787	0.7787	0.4309	0.4308	0.7787	0.5	0.5
25.9	415615.4727	0.7789	0.7789	0.4317	0.4317	0.7789	0.5	0.5

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
25.95	416460.937	0.779	0.779	0.4325	0.4325	0.779	0.5	0.5
26.0	417342.1015	0.7792	0.7792	0.4334	0.4333	0.7792	0.5	0.5
26.05	418258.4731	0.7793	0.7793	0.4342	0.4342	0.7793	0.5	0.5
26.1	419104.8075	0.7795	0.7795	0.435	0.435	0.7795	0.5	0.5
26.15	419986.8786	0.7796	0.7796	0.4359	0.4358	0.7796	0.5	0.5
26.2	420904.1928	0.7798	0.7798	0.4367	0.4367	0.7798	0.5	0.5
26.25	421751.3974	0.7799	0.7799	0.4375	0.4375	0.7799	0.5	0.5
26.3	422634.3751	0.7801	0.78	0.4384	0.4383	0.78	0.5	0.5
26.35	423552.6318	0.7802	0.7802	0.4392	0.4392	0.7802	0.5	0.5
26.4	424400.7066	0.7803	0.7803	0.44	0.44	0.7803	0.5	0.5
26.45	425284.591	0.7805	0.7805	0.4409	0.4408	0.7805	0.5	0.5
26.5	426203.7901	0.7806	0.7806	0.4417	0.4417	0.7806	0.5	0.5
26.55	427052.7351	0.7808	0.7808	0.4425	0.4425	0.7808	0.5	0.5
26.6	427937.5261	0.7809	0.7809	0.4434	0.4433	0.7809	0.5	0.5
26.65	428857.6678	0.7811	0.7811	0.4442	0.4442	0.7811	0.5	0.5
26.7	429707.483	0.7812	0.7812	0.445	0.445	0.7812	0.5	0.5
26.75	430593.1806	0.7814	0.7814	0.4459	0.4458	0.7814	0.5	0.5
26.8	431516.1127	0.7815	0.7815	0.4467	0.4467	0.7815	0.5	0.5
26.85	432411.2293	0.7817	0.7817	0.4475	0.4475	0.7817	0.5	0.5
26.9	433342.3383	0.782	0.782	0.4484	0.4483	0.782	0.5	0.5
26.95	434213.9048	0.782	0.782	0.4492	0.4492	0.782	0.5	0.5
27.0	435016.9045	0.7821	0.7821	0.45	0.45	0.7821	0.5	0.5
27.05	435663.2817	0.7818	0.7818	0.4509	0.4508	0.7818	0.5	0.5
27.1	436340.3573	0.7815	0.7815	0.4517	0.4517	0.7815	0.5	0.5
27.15	437097.4037	0.7815	0.7815	0.4525	0.4525	0.7815	0.5	0.5
27.2	437886.092	0.7815	0.7815	0.4534	0.4533	0.7815	0.5	0.5
27.25	438705.9423	0.7814	0.7814	0.4542	0.4542	0.7814	0.5	0.5
27.3	439462.8146	0.7814	0.7814	0.455	0.455	0.7814	0.5	0.5
27.35	440251.3217	0.7814	0.7814	0.4559	0.4558	0.7814	0.5	0.5
27.4	441070.9834	0.7813	0.7813	0.4567	0.4567	0.7813	0.5	0.5
27.45	441827.6817	0.7813	0.7813	0.4575	0.4575	0.7813	0.5	0.5
27.5	442616.0074	0.7813	0.7813	0.4584	0.4583	0.7813	0.5	0.5
27.55	443435.4806	0.7812	0.7812	0.4592	0.4592	0.7812	0.5	0.5
27.6	444192.0049	0.7812	0.7812	0.46	0.46	0.7812	0.5	0.5
27.65	444980.1493	0.7812	0.7812	0.4609	0.4608	0.7812	0.5	0.5
27.7	445799.434	0.7812	0.7812	0.4617	0.4617	0.7812	0.5	0.5

$Q_{tot} \left(\frac{m^3}{s} \right)$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
27.75	446555.7842	0.7811	0.7811	0.4625	0.4625	0.7811	0.5	0.5
27.8	447343.7473	0.7811	0.7811	0.4634	0.4633	0.7811	0.5	0.5
27.85	448162.8435	0.7811	0.7811	0.4642	0.4642	0.7811	0.5	0.5
27.9	448919.0197	0.781	0.781	0.465	0.465	0.781	0.5	0.5
27.95	449706.8015	0.781	0.781	0.4659	0.4658	0.781	0.5	0.5
28.0	450525.7092	0.781	0.781	0.4667	0.4667	0.781	0.5	0.5
28.05	451281.7113	0.781	0.781	0.4675	0.4675	0.781	0.5	0.5
28.1	452069.3117	0.7809	0.7809	0.4684	0.4683	0.7809	0.5	0.5
28.15	452888.0309	0.7809	0.7809	0.4692	0.4692	0.7809	0.5	0.5
28.2	453643.859	0.7809	0.7809	0.47	0.47	0.7809	0.5	0.5
28.25	454431.2782	0.7808	0.7808	0.4709	0.4708	0.7808	0.5	0.5
28.3	455249.8088	0.7808	0.7808	0.4717	0.4717	0.7808	0.5	0.5
28.35	456005.4629	0.7808	0.7808	0.4725	0.4725	0.7808	0.5	0.5
28.4	456792.7007	0.7807	0.7808	0.4734	0.4733	0.7808	0.5	0.5
28.45	457611.0429	0.7807	0.7807	0.4742	0.4742	0.7807	0.5	0.5
28.5	458366.5229	0.7807	0.7807	0.475	0.475	0.7807	0.5	0.5
28.55	459153.5794	0.7807	0.7807	0.4759	0.4758	0.7807	0.5	0.5
28.6	459971.7331	0.7806	0.7806	0.4767	0.4767	0.7806	0.5	0.5
28.65	460727.0391	0.7806	0.7806	0.4775	0.4775	0.7806	0.5	0.5
28.7	461513.9142	0.7806	0.7806	0.4784	0.4783	0.7806	0.5	0.5
28.75	462331.8794	0.7805	0.7805	0.4792	0.4792	0.7805	0.5	0.5
28.8	463087.0113	0.7805	0.7805	0.48	0.48	0.7805	0.5	0.5
28.85	463873.7052	0.7805	0.7805	0.4809	0.4808	0.7805	0.5	0.5
28.9	464691.4818	0.7805	0.7805	0.4817	0.4817	0.7805	0.5	0.5
28.95	465446.4398	0.7804	0.7804	0.4825	0.4825	0.7804	0.5	0.5
29.0	466232.9522	0.7804	0.7804	0.4834	0.4833	0.7804	0.5	0.5
29.05	467050.5404	0.7804	0.7804	0.4842	0.4842	0.7804	0.5	0.5
29.1	467805.3243	0.7803	0.7803	0.485	0.485	0.7803	0.5	0.5
29.15	468591.6555	0.7803	0.7803	0.4859	0.4858	0.7803	0.5	0.5
29.2	469409.0551	0.7803	0.7803	0.4867	0.4867	0.7803	0.5	0.5
29.25	470163.665	0.7803	0.7803	0.4875	0.4875	0.7803	0.5	0.5
29.3	470949.8148	0.7802	0.7802	0.4884	0.4883	0.7802	0.5	0.5
29.35	471767.026	0.7802	0.7802	0.4892	0.4892	0.7802	0.5	0.5
29.4	472521.4618	0.7802	0.7802	0.49	0.49	0.7802	0.5	0.5
29.45	473307.4303	0.7801	0.7801	0.4909	0.4908	0.7801	0.5	0.5
29.5	474124.453	0.7801	0.7801	0.4917	0.4917	0.7801	0.5	0.5

$Q_{tot} (\frac{m^3}{s})$	$P_{tot} (W)$	$\eta_1 (-)$	$\eta_2 (-)$	$q_0 (-)$	$q_1 (-)$	$\eta_{tot} (-)$	$r_0 (-)$	$r_1 (-)$
29.55	474878.7147	0.7801	0.7801	0.4925	0.4925	0.7801	0.5	0.5
29.6	475664.5019	0.78	0.78	0.4934	0.4933	0.78	0.5	0.5
29.65	476481.3361	0.78	0.78	0.4942	0.4942	0.78	0.5	0.5
29.7	477235.4238	0.78	0.78	0.495	0.495	0.78	0.5	0.5
29.75	478021.0297	0.78	0.78	0.4959	0.4958	0.78	0.5	0.5
29.8	478837.6753	0.7799	0.7799	0.4967	0.4967	0.7799	0.5	0.5
29.85	479591.589	0.7799	0.7799	0.4975	0.4975	0.7799	0.5	0.5
29.9	480377.0136	0.7799	0.7799	0.4984	0.4983	0.7799	0.5	0.5
29.95	481193.4707	0.7798	0.7798	0.4992	0.4992	0.7798	0.5	0.5
30.0	481947.2104	0.7798	0.7798	0.5	0.5	0.7798	0.5	0.5

Table A.3: Output file for basic example

A.2 Four unit example

A.2.1 Input files

The four unit example used a lower discretization, therefore the efficiency tables also had a lower discretization. Underneath are the input files for generator and transformer attached, as well as for the four different hydraulic machines.

Relative Power	Efficiency
0.0	0.886
0.1	0.932
0.2	0.96
0.3	0.979
0.4	0.986
0.5	0.9866
0.6	0.9879
0.7	0.9888
0.8	0.9893
0.9	0.9896
1.0	0.9898
1.109	0.9915

Table A.4: Generator input file for all units

n (rpm)	Q (m ³ /s)	η (%)	H (m)
187.5	21.1074	0.0016	119.0
187.5	30.915	0.3412	119.0
187.5	42.8005	0.5636	119.0
187.5	55.1577	0.6853	119.0
187.5	68.6911	0.7772	119.0
187.5	82.7798	0.8446	119.0
187.5	97.0035	0.8948	119.0
187.5	111.3193	0.9271	119.0
187.5	113.2619	0.9294	119.0
187.5	115.2548	0.9312	119.0
187.5	117.3523	0.9329	119.0
187.5	119.2807	0.9329	119.0
187.5	121.2895	0.9341	119.0
187.5	122.4947	0.9356	119.0
187.5	123.5393	0.9343	119.0
187.5	124.6117	0.9333	119.0
187.5	125.8293	0.9351	119.0
187.5	127.4296	0.9399	119.0
187.5	128.6878	0.9429	119.0
187.5	130.1225	0.9426	119.0
187.5	130.7656	0.9436	119.0
187.5	137.3596	0.9497	119.0
187.5	150.1756	0.965	119.0
187.5	161.5806	0.9607	119.0
187.5	171.2357	0.9561	119.0
187.5	180.9942	0.9452	119.0
187.5	189.568	0.9337	119.0

Table A.5: Hydraulic machines input file unit 1

n (rpm)	Q (m ³ /s)	η (%)	H (m)
187.5	21.1074	0.0015	1.0
187.5	30.915	0.3243	1.0
187.5	42.8005	0.5357	1.0
187.5	55.1577	0.6513	1.0
187.5	68.6911	0.7387	1.0
187.5	82.7798	0.8027	1.0
187.5	97.0035	0.8504	1.0
187.5	111.3193	0.8812	1.0
187.5	113.2619	0.8833	1.0
187.5	115.2548	0.885	1.0
187.5	117.3523	0.8866	1.0
187.5	119.2807	0.8867	1.0
187.5	121.2895	0.8877	1.0
187.5	122.4947	0.8892	1.0
187.5	123.5393	0.888	1.0
187.5	124.6117	0.887	1.0
187.5	125.8293	0.8887	1.0
187.5	127.4296	0.8933	1.0
187.5	128.6878	0.8961	1.0
187.5	130.1225	0.8959	1.0
187.5	130.7656	0.8968	1.0
187.5	137.3596	0.9026	1.0
187.5	150.1756	0.9171	1.0
187.5	161.5806	0.913	1.0
187.5	171.2357	0.9086	1.0
187.5	180.9942	0.8983	1.0
187.5	189.568	0.8873	1.0

Table A.6: Hydraulic machines input file unit 2

n (rpm)	Q (m ³ /s)	η (%)	H (m)		
187.5	21.1074	0.3344	1.0	nan	nan
187.5	30.915	0.5523	1.0	nan	9.8075
187.5	42.8005	0.6716	1.0	nan	11.8856
187.5	55.1577	0.7617	1.0	nan	12.3571
187.5	68.6911	0.8277	1.0	nan	13.5334
187.5	82.7798	0.8769	1.0	nan	14.0887
187.5	97.0035	0.9086	1.0	nan	14.2237
187.5	111.3193	0.9109	1.0	nan	14.3158
187.5	113.2619	0.9126	1.0	nan	1.9426
187.5	115.2548	0.9142	1.0	nan	1.9929
187.5	117.3523	0.9143	1.0	nan	2.0975
187.5	119.2807	0.9154	1.0	nan	1.9284
187.5	121.2895	0.9169	1.0	nan	2.0088
187.5	122.4947	0.9156	1.0	nan	1.2053
187.5	123.5393	0.9147	1.0	nan	1.0446
187.5	124.6117	0.9164	1.0	nan	1.0724
187.5	125.8293	0.9211	1.0	nan	1.2176
187.5	127.4296	0.924	1.0	nan	1.6004
187.5	128.6878	0.9238	1.0	nan	1.2582
187.5	130.1225	0.9247	1.0	nan	1.4347
187.5	130.7656	0.9307	1.0	nan	0.6431
187.5	137.3596	0.9457	1.0	nan	6.594
187.5	150.1756	0.9414	1.0	nan	12.8159
187.5	161.5806	0.9369	1.0	nan	11.405
187.5	171.2357	0.9263	1.0	nan	9.6551
187.5	180.9942	0.915	1.0	nan	9.7585
187.5	189.568	0.8987	1.0	nan	8.5738

Table A.7: Hydraulic machines input file unit 3

n (rpm)	Q (m ³ /s)	η (%)	H (m)
187.5	21.1074	0.0016	1.0
187.5	30.915	0.3412	1.0
187.5	42.8005	0.5636	1.0
187.5	55.1577	0.6853	1.0
187.5	68.6911	0.7772	1.0
187.5	82.7798	0.8446	1.0
187.5	97.0035	0.8948	1.0
187.5	111.3193	0.9271	1.0
187.5	113.2619	0.9294	1.0
187.5	115.2548	0.9312	1.0
187.5	117.3523	0.9329	1.0
187.5	119.2807	0.9329	1.0
187.5	121.2895	0.9341	1.0
187.5	122.4947	0.9356	1.0
187.5	123.5393	0.9343	1.0
187.5	124.6117	0.9333	1.0
187.5	125.8293	0.9351	1.0
187.5	127.4296	0.9399	1.0
187.5	128.6878	0.9429	1.0
187.5	130.1225	0.9426	1.0
187.5	130.7656	0.9436	1.0
187.5	137.3596	0.9497	1.0
187.5	150.1756	0.965	1.0
187.5	161.5806	0.9607	1.0
187.5	171.2357	0.9561	1.0
187.5	180.9942	0.9452	1.0
187.5	189.568	0.9337	1.0

Table A.8: Hydraulic machines input file unit 4

Relative Power	Efficiency
0.0	0.886
0.1	0.932
0.2	0.96
0.3	0.979
0.4	0.986
0.5	0.9866
0.6	0.9879
0.7	0.9888
0.8	0.9893
0.9	0.9896
1.0	0.9898
1.109	0.9915

Table A.9: Transformer input file for all units